

Spatio-Temporal Trajectory Similarity Measures: A Comprehensive Survey and Quantitative Study

Danlei Hu, Lu Chen, Hanxi Fang, Ziquan Fang, Yunjun Gao, *Member, IEEE*

Abstract—Spatio-temporal trajectory data analytics has inspired a broad range of real-life applications across different fields. Trajectory similarity measure, which aims to evaluate the distance between two trajectories, is a fundamental functionality of trajectory analytics. In this paper, we propose a comprehensive survey that summarizes almost all the representative spatio-temporal trajectory measures in terms of three hierarchical perspectives (i.e., Non-learning vs. Learning, Free Space vs. Road Network, and Standalone vs. Distributed). Moreover, we provide an evaluation benchmark (publicly available at <https://github.com/ZJU-DAILY/TSM>) by designing five real-world transformation scenarios. Based on this benchmark, extensive experiments are conducted to study the effectiveness, robustness, efficiency, and scalability of each measure, which offers a reference for trajectory measure selection among traditional similarity measures, deep learning models, and distributed processing technologies.

Index Terms—Trajectory Similarity Measure, Distributed Similarity Search, Deep Representation Learning, Experimental Evaluation

1 INTRODUCTION

WITH the proliferation of GPS-equipped devices and mobile computing services, massive spatio-temporal trajectory data of moving objects such as people, vessels, and vehicles are being captured [41], [89]. For example, people share visited places (e.g., POIs) on social networks by their smartphones to generate “check-in” trajectories; according to the AIS project [49], millions of vessels connected with the AIS services continuously report locations to ensure the sailing safety; the world’s largest ridesharing company Uber collects up to 17 million vehicle trips daily [65].

Trajectory data with its analytics benefit a broad range of real-life applications across different fields such as urban computing [40], transportation [90], behavior study [38], and public security [35], to name but a few. A fundamental functionality of most trajectory analysis is to evaluate the relationship/distance between two trajectories, i.e., trajectory similarity measurement. With an accurate and efficient trajectory similarity measure, downstream trajectory analytics involving retrieval [52], [60], clustering [18], [33], classification [22], [32], and mobility pattern mining [29], [46], [62] tasks can be well-supported to serve upper applications. For instance, Xie et al. [73] propose two distance measures to support similarity queries and joins in a large-scale trajectory dataset. Wang et al. [69] design a road network oriented distance measure for vehicle trajectory similarity computation, based on which, k -means clustering is studied [71] to detect hot/popular traveling paths in a city. In both cases, the trajectory similarity measure plays a fundamental role, and a different measure selection may result in totally different query results and clustering quality. Take the trajectory clustering as an example, clustering aims to group similar trajectories into clusters, where similarity computation is a fundamental task of clustering as shown

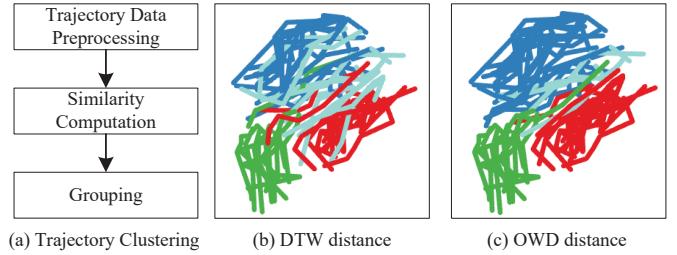


Fig. 1. Example of Trajectory Clustering

in Figure 1(a). Different similarity measures may affect the clustering results. As shown in Figures 1(b)–1(c), the clustering results are quite different when applying different measures (i.e., DTW and OWD). Hence, the selection of similarity measure is important. In addition, more trajectory similarity based analyses tasks can refer to [41], [68].

Unlike isolated spatial points or one-dimensional time series where the distance definition is straightforward, it is non-trivial to define the distance between continuous two-dimensional trajectories. It also needs to consider following four trajectory characteristics: (i) Different data sources, i.e., free space vs. road network space. In the latter case, a proper trajectory measure should take the road topology into account, as people and vehicles cannot travel like vessels without spatial constraints [56]. (ii) Various sampling rates and lengths. Unlike time series that generally feature constant and high sampling rates [37], spatio-temporal trajectory data is usually generated via varying samplings, resulting in variable lengths. (iii) The effect of noise. The noise points commonly exist, especially due to strength attenuation and interference in urban cities [85]. (iv) Complex shapes. Compared to private-car trajectories that are usually inaccessible caused by privacy principles, taxi trajectories are widely studied in the community [15], [63], [70]. However, taxi trajectories exhibit much more diverse, complex, and flexible geometric, because of various pick-up demands. To deal with the above spatio-temporal characteristics, tremendous amounts of research efforts are devoted to design dozens of

• D. Hu, L. Chen (Corresponding Author), Z. Fang and Y. Gao are with the College of Computer Science, Zhejiang University, Hangzhou 310027, China, E-mail:{dlhu, luchen, zafang, gaoyj}@zju.edu.cn.
• H. Fang is with the School of Earth Science, Zhejiang University, Hangzhou 310027, China, E-mail:hanxif@zju.edu.cn.

TABLE 1

Spatio-Temporal Trajectory Similarity Measures. (Here, m and n represent the lengths of two trajectories, d denotes the dimension of embedded vector, and the dark backdrop indicates the studied scope in the state-of-the-art survey [59])

Category		Measure	Complexity	Metric	Unequal Length	Parameter Free	Noise Sensitive	Representation
Non-learning	Free Space	Standalone	ED	$O(n)$	✓	✗	✗	Point
			DTW [81]	$O(mn)$	✗	✓	✗	Point
			LCSS [66]	$O(mn)$	✗	✓	✓	Point
			EDR [13]	$O(mn)$	✗	✓	✗	Point
			EDwP [51]	$O(mn)$	✗	✓	✓	Segment
			ERP [12]	$O(mn)$	✓	✓	✗	Point
			Hausdorff [4]	$O(mn)$	✓	✓	✓	Point
			Frechet [2]	$O(mn)$	✗	✓	✗	Point
			LIP [48]	$O((m + n)\log(m + n))$	✗	✓	✓	Point
			OWD [21]	$O(mn)$	✗	✓	✗	Point
		Distributed	Seg-Frechet [73]	$O(mn)$	✗	✓	✓	Segment
			DFT [73]	$O(mn)$	✗	✓	✓	Segment
			DITA [57]	$O(mn)$	✗	✓	✗	Point
	Road Network	Standalone	REPOSE [87]	$O(mn)$	✗	✓	✗	Point
			NetDTW [74]	$O(mn)$	✗	✓	✓	Point
			NetLCSS [19]	$O(mn)$	✗	✓	✓	Point
			LORS [69]	$O(mn)$	✗	✓	✓	Segment
			TP [56]	$O(mn)$	✗	✓	✗	Point
			NetERP [31]	$O(mn)$	✓	✓	✗	Point
			NetEDR [31]	$O(mn)$	✗	✓	✓	Point
			LCRS [82]	$O(mn)$	✗	✓	✓	Segment
		Distributed	DISON [82]	$O(mn)$	✗	✓	✗	Segment
Learning	Free Space	Standalone	NEUTRAJ [78]	$O(m + n)$	/	✓	✗	Vector
			Traj2SimVec [86]	$O(m + n)$	/	✓	✗	Vector
	Road Network	Standalone	GTS [27]	$O(d)$	/	✓	✗	Vector
			ST2Vec [19]	$O(d)$	/	✓	✗	Vector

spatio-temporal trajectory similarity measures in the literature.

Being faced with a huge amount of trajectory measures, researchers are often too exhausted to select a proper one. On the one hand, there are too many trajectory measures, which were proposed under different scenarios, e.g., learning based or non-learning based, free-space oriented or road network oriented, as well as standalone or distributed processing. In each scenario, various measures also exist. Consequently, users need to spend tons of time and efforts to explore the specific details of each measure and the relations/differences among them. On the other hand, the evaluations on various trajectory measures are still not well organized. For instance, some measures only focus on efficiency, while others may put more attention on effectiveness and robustness. To address the problems mentioned above, a comprehensive survey, benchmark, and evaluation will be a great help for researchers involved in this important topic. Specifically, considering three-dimensional aspects, we classify the existing representative spatio-temporal trajectory measures proposed from 1995 to 2022 year in a hierarchical way, i.e., **Non-learning** vs. **Learning** (first hierarchy), **Free Space** vs. **Road Network** (second hierarchy), and **Standalone** vs. **Distributed** (third hierarchy). Table 1 summarizes trajectory similarity measures.

Although previous systematic surveys have made some efforts, they mainly focus on non-learning, free-space, or standalone based measures and thus significantly narrow the studied scope of trajectory similarity community. For example, Gudmundsson and Toohey et al. [64] only review four basic trajectory distance measures including Euclidean distance (ED), DTW [81], LCSS [66], and Fréchet [2]. Based on it, Sousa et al. [15] append partial vehicle trajectory similarity measures, e.g., LORS [69], LCRS [82], and Net-

EDR [31]. Note that, experimental evaluation of these measures are not studied in these surveys. Su et al. [59] conduct the state-of-the-art survey and experimental evaluation for 10 trajectory similarity measures under the non-learning, free-space, and standalone contexts. In contrast, we conduct a much more systematic review and evaluation of almost all representative spatio-temporal trajectory measures in a three-dimensional hierarchical way as shown in Table 1. We conduct a three-dimensional survey due to three following motivations:

(i) As deep learning has made great success in AI community, many researchers [19], [27], [37], [78], [86] start leveraging the powerful approximation capabilities of neural networks to attempt to replace the traditional handcrafted trajectory measures with learning-based models. Besides, increasing efforts have been devoted to embracing learning-based techniques as an integral part of trajectory data management and analytics, such as deep clustering [20], deep mobility pattern mining [9], and deep path recommendation [83], to name just a few. As such, we believe that we have reached an imperative point to systematically study the contribution and explore the relations, differences, and pros/cons among the emerging learning-based and classic non-learning based trajectory measures.

(ii) In the early stage, most of the trajectory measures [59] are proposed for objects that move freely in the Euclidean space, e.g., bird or vessel trajectories. Recently, the proliferation of vehicle navigation systems and location based services (LBSs) enable the massive collection of vehicle and people trajectories in road networks. In that case, the free-space oriented trajectory measures cannot reflect the true distance between moving objects in a moving-constrained road network. In view of this, many network-aware trajectory measures [31], [56], [69], [82] are designed. To the best

of our knowledge, there is no previous work that has given a systematic review and experimental evaluation of these network-based trajectory distance measures.

(iii) Since the storage capacity and processing ability of a single machine can no longer support a large scale of trajectory data, another popular line of trajectory similarity study is designing efficient and scalable frameworks upon distributed processing platforms (e.g., Spark) for large-scale trajectory similarity analytics. Towards this, several system-level frameworks are also developed. Hence, we are inspired to present a sufficient review and performance evaluation of distributed-based similarity computation studies.

Motivated by the observations above, we conduct a most unprecedented survey on almost all the representative spatio-temporal trajectory similarity measures proposed in the literature. Specifically, we review 25 similarity measures from the following hierarchical perspectives, namely: (i) Model architecture, i.e., non-learning based vs. learning-based; (ii) Space context, i.e., free space oriented vs. road network oriented; (iii) Computational mechanism, i.e., standalone machine vs. distributed platform. To the best of our knowledge, none of the previous studies have reached such a wide scope of trajectory measure evaluations. It is also significant to conduct objective and sufficient evaluations to study each measure by using the same tasks, datasets, and experimental settings. In light of this, we provide a standard benchmark in a real setting by introducing five different scenarios, i.e., length shift, shape shift, noise shift, sampling shift, and cardinality shift, on two real datasets. Then, we extensively study and compare the effectiveness, robustness, efficiency, and scalability for all measures via quantitative and qualitative analysis. With our benchmark, (i) users can quickly grasp the technical details about trajectory similarity measures; and (ii) users can easily choose or design suitable measures and use them as the baselines. Overall, this paper makes the contributions below.

- We conduct a concise but concrete review to evaluate and compare spatio-temporal trajectory similarity measures qualitatively and quantitatively in three dimensions, i.e., learning-based or not, road network oriented or not, and distributed-based or not, providing a reference for measure selection among traditional similarity measures, deep learning models, and distributed processing technologies.
- We provide a standard evaluation benchmark with five types of trajectory transformations under five typical trajectory analytics scenarios. Based on it, we conduct a comprehensive evaluation of effectiveness, robustness, efficiency, and scalability performance of 25 representative measures.
- We have several key observations according to experimental results, based on which, we offer insights about trajectory measure selection in specific application scenarios. According to several issues that remained to be solved, we also present detailed potential future directions.

Section 2 gives the preliminaries. Sections 3 and 4 present the non-learning-based and learning-based trajectory similarity measures respectively, where free-space vs. network oriented and standalone vs. distributed measures are further

detailed. Section 5 provides the benchmark. The evaluation results with insights are reported in Section 6. Finally, Section 8 concludes the paper.

2 PRELIMINARIES

In this section, we provide the definitions related to trajectory similarity measure.

Definition 1. (Trajectory) A trajectory T is an ordered sequence that consists of GPS sampling points, i.e., $T = \{p_1, p_2, \dots, p_n\}$, where $p_i (1 \leq i \leq n)$ is an observed GPS location in d -dimensional tuple to describe the mobility of T .

Here, n denotes the length of a trajectory. We use T_i to denote a trajectory whose id is i , and use T_i^j to denote the j -th point in T_i . The points observed by GPS equipment could contain various information. For simplicity, we assume each sampling point is of two-dimension or three-dimension (i.e., $(latitude, longitude)$ or $(latitude, longitude, timestamp)$). Although trajectory data is similar to time series data, time series data typically lacks of spatial information. Especially for vehicle trajectories, they are physically constrained to the road network. Thus, directly applying similarity measures for time series in free space is not always suitable. Hence, trajectories with road network constraints are taken into consideration. The road network is defined as follows.

Definition 2. (Road Network) A road network is represented as a directed graph $G = (V, E)$, where V is a set of road intersections (i.e., road vertices) in the road network, and E is a set of edges with direction of road segments.

Here, $v_i = (lat_i, lon_i) \in V$ is a road intersection, where i is the id of v_i , while (lat_i, lon_i) denotes the latitude and longitude of v_i . Meanwhile, an edge $e_{i,j} \in E$ represents a directed road segment from v_i to v_j . Existing map matching methods [7], [84] are proposed to transform a trajectory of GPS points into a road network constrained trajectory. A trajectory in road network can be represented as $T = \langle v_1, v_2, \dots, v_k \rangle (k \leq n)$ or $T = \langle e_1, e_2, \dots, e_{k-1} \rangle (k \leq n)$, where k is the number of road vertices/edges in T .

Definition 3. (Similarity Measure) A similarity measure is a function $f(T, Q)$ to measure the distance between T and Q .

Definition 4. (Metric Measure) Given a similarity measure f and any three trajectories T_i , T_j and T_k , we call f as a metric measure if f satisfies the following conditions: (i) Uniqueness: $f(T_i, T_j) = 0 \leftrightarrow T_i = T_j$; (ii) Nonnegativity: $f(T_i, T_j) \leq 0$; (iii) Triangle Inequality: $f(T_i, T_k) \leq f(T_i, T_j) + f(T_j, T_k)$; (iv) Symmetry: $f(T_i, T_j) = f(T_j, T_i)$.

As shown in Table 1, a similarity measure can be classified according to i) whether it is metric or not as defined in Definition 4 ("✓" denotes it is a metric, "✗" denotes it is not a metric, "/" denotes learning methods do not use a manual function to compute similarity), ii) whether it can support trajectories with different lengths, iii) whether it is parameter-free, and iv) whether it is noise sensitive. Since trajectory similarity search is a prerequisite task in upper-level applications [25], [73], we perform top- k trajectory similarity search to evaluate the capabilities of all similarity measures in Section 6, which is defined below.

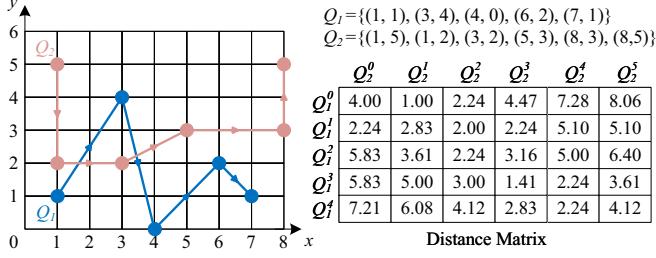


Fig. 2. Running Example of Trajectory Computation

Definition 5. (Top- k Similarity Search) Given a query trajectory Q , a set of trajectories $\mathcal{T} = \{T_1, T_2, \dots, T_{|\mathcal{T}|}\}$, and a similarity measure $f(\cdot)$, the top- k similarity search returns k trajectories in \mathcal{T} that are most similar to Q ($S = \{S \subseteq \mathcal{T} \wedge |S| = k \wedge \forall T_s \in S, \forall T_o \in O - S (f(Q, T_s) \geq f(Q, T_o))\}$). Here, “ \geq ” means “ \geq ” for LCSS while denotes “ \leq ” for others.

3 NON-LEARNING BASED MEASURES

In this section, we introduce 22 non-learning based measures (as shown in the upper part of Table 1), and further detail each measure in terms of space context (i.e., free space and road network).

3.1 Trajectory Similarity in Free Space

We review 14 free space measures including 3 measures in the distributed settings. Each measure is point-based or segment-based. Here, we provide an running example in Figure 2, which depicts two trajectories Q_1 and Q_2 with the corresponding distance matrix to store the distances between points of two trajectories.

3.1.1 Point-based Measures.

Point-based measures are widely utilized to compute the similarity among trajectories consisted of sampling points.

Euclidean Distance (ED). Euclidean distance is a well-known measure, which is used to calculate the distance between time series with the same length [30]. When applied to trajectory data, sampling points in two series are aligned in order [55]. Formally, ED between two trajectories T_1 and T_2 is defined as:

$$d(p_1, p_2) = \sqrt{(p_1.lat - p_2.lat)^2 + (p_1.lon - p_2.lon)^2} \quad (1)$$

$$ED(T_1, T_2) = \frac{\sum_{i=1}^N d(T_1^i, T_2^i)}{N} \quad (2)$$

Here, $d(p_1, p_2)$ denotes the distance between two points p_1 and p_2 , lat and lon denote the latitude and longitude of the point, while N denotes the length of T_1 and T_2 . ED is a metric and parameter-free measure with the time complexity of $O(N)$. However, it assumes that trajectories are with the same length, while it is not realistic in real life. Thus, we do not focus on evaluating the performance of ED. Note that, in Figure 2, Q_1 and Q_2 are with different lengths, and thus, ED cannot be applied to compute the similarity in this running example.

Dynamic Time Warping Distance (DTW). Similar to ED, DTW is also a measure designed for time series and can be applied to trajectory data [23], [45]. However, different from ED, DTW can align a point of one trajectory to one or more consecutive points of another trajectory (see Figure 3).

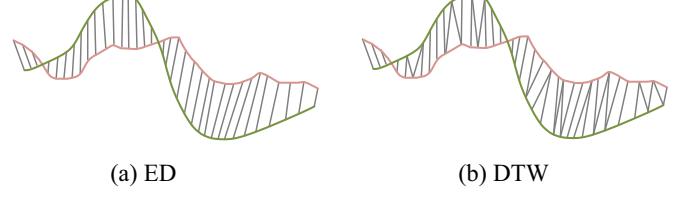


Fig. 3. ED and DTW

The time complexity of DTW is $O(mn)$ where m and n denote the length of two comparing trajectories respectively. Although DTW is non-metric, it is most commonly used for trajectory similarity computation. Various techniques are proposed to improve the efficiency of DTW [30], [54]. DTW distance between two trajectories T_1 and T_2 is defined as:

$$DTW(T_1, T_2) = \begin{cases} 0 & \text{if } m = n = 0 \\ \infty & \text{if } m = 0 \text{ or } n = 0 \\ d(T_1^m, T_2^n) + \\ \min \begin{cases} DTW(Head(T_1), T_2), \\ DTW(T_1, Head(T_2)), \\ DTW(Head(T_1), Head(T_2)) \end{cases} & \text{otherwise} \end{cases} \quad (3)$$

Where $Head(\cdot)$ is to get the head sampling points except the last one point (i.e., $Head(T) = \{p_1, p_2, \dots, p_{n-1}\}$), and $d(\cdot, \cdot)$ denotes the distance between two points. Take Q_1 and Q_2 in Figure 2 as example, we are able to compute the DTW distance matrix in Figure 4(a) and have $DTW(Q_1, Q_2) = 16.52$.

Longest Common Subsequence (LCSS). LCSS distance [53], [66] between two trajectories T_1 and T_2 is defined as the size of the longest common subsequence of T_1 and T_2 . Given a threshold parameter ϵ to determine whether a pair of points (p_1, p_2) matches, if $d(p_1, p_2) \leq \epsilon$, (p_1, p_2) is a match pair. As shown in Figure 4(b), when $\epsilon = 1.0$, the pair (Q_1^0, Q_2^1) is a match pair, as $d(Q_1^0, Q_2^1) \leq \epsilon$. Actually, LCSS distance finds all match pairs in two trajectories, and it is non-metric but noise sensitive compared to ED. Moreover, its time complexity is $O(mn)$. Formally, LCSS distance is defined as:

$$LCSS_\epsilon(T_1, T_2) = \begin{cases} 0 & \text{if } m = 0 \text{ or } n = 0 \\ 1 + LCSS_\epsilon(Head(T_m), Head(T_n)) & \text{if } d(T_1^m, T_2^n) \leq \epsilon \\ \max \begin{cases} LCSS_\epsilon(Head(T_1), T_2), \\ LCSS_\epsilon(T_1, Head(T_2)) \end{cases} & \text{otherwise} \end{cases} \quad (4)$$

As shown in Figure 4(b), when $\epsilon = 1.0$, $LCSS(Q_1, Q_2) = 1$.

Edit Distance on Real Sequence (EDR). Similar to LCSS, EDR distance [1], [13], [16], [33] is also one of the edit distance-based measures with the time complexity of $O(mn)$, where edit distance equals to the number of necessary operations (i.e., insertion, deletion and replacement) when transforming one trajectory into another one. However, different from LCSS, EDR sets a parameter $cost$ to measure the cost of an unmatched pair of points. Formally, EDR is defined as:

$$EDR(T_1, T_2) = \begin{cases} n & \text{if } m = 0 \\ m & \text{if } n = 0 \\ \min \begin{cases} EDR(Head(T_1), T_2) + 1, \\ EDR(T_1, Head(T_2)) + 1, \\ EDR(Head(T_1), Head(T_2)) + cost \end{cases} & \text{otherwise} \end{cases} \quad (5)$$

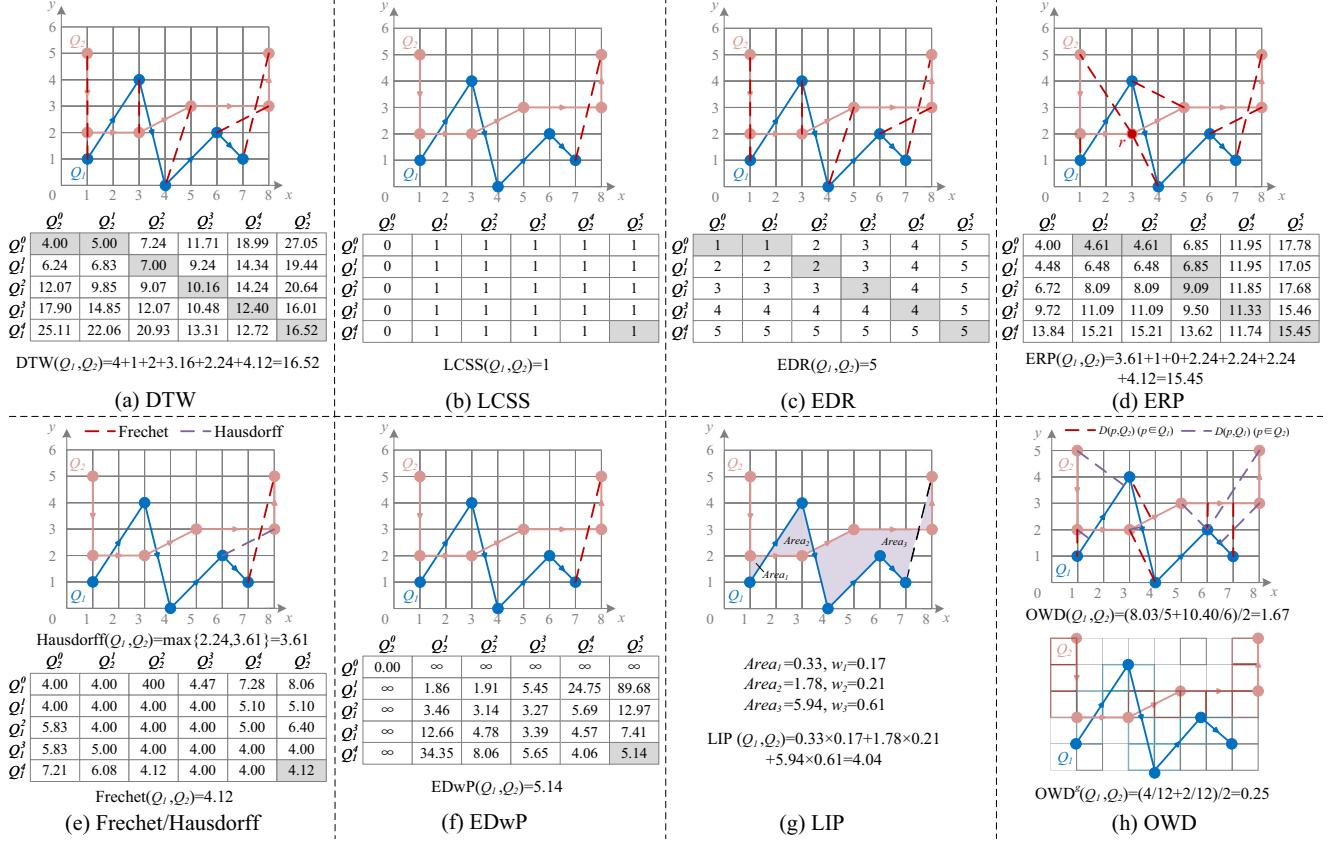


Fig. 4. Running Examples on Non-learning Based Measures in Free Space

Here, if $d(T_1^m, T_2^n) \leq \epsilon$, then $\text{cost} = 0$; otherwise $\text{cost} = 1$. As depicted in Figure 4(c), if $\epsilon = 1.0$, $\text{EDR}(Q_1, Q_2) = 5$.

Edit Distance with Real Penalty (ERP). The above edit distance-based measures are basically non-metric. However, ERP [12], [14], [92] is a metric measure that can be used for indexing and pruning. Different from other edit distance-based measures such as LCSS and EDR, ERP does not require a threshold parameter, but sets a reference point (i.e., gap point) for measuring. ERP is commonly used in trajectory similarity computation with the time complexity of $O(mn)$ because of its metricity. Formally, ERP distance between two trajectories T_1 and T_2 is defined as:

$$\text{ERP}(T_1, T_2) = \begin{cases} \sum_{i=1}^n d(T_2^i, r) & \text{if } m = 0 \\ \sum_{i=1}^m d(T_1^i, r) & \text{if } n = 0 \\ \min \left\{ \begin{array}{l} \text{ERP}(\text{Head}(T_1), \text{Head}(T_2)) + d(T_1^m, T_2^n), \\ \text{ERP}(T_1, \text{Head}(T_2)) + d(T_2^n, r), \\ \text{ERP}(\text{Head}(T_1), T_2) + d(T_1^m, r) \end{array} \right. & \text{otherwise} \end{cases} \quad (6)$$

Where r denotes the reference point. Considering the running examples shown in Figure 2, when we set the reference point r as $(3, 2)$ (see Figure 4(d)), $\text{ERP}(Q_1, Q_2) = 15.45$.

Hausdorff Distance. Hausdorff distance [4] is a metric measure, which measures the maximum distance of all distance values from a point of one trajectory to the nearest point in another trajectory. The time complexity of Hausdorff distance is $O(mn)$, which is a parameter-free measure. Formally, Hausdorff distance is defined as:

$$\text{Hausdorff}(T_1, T_2) = \max \left\{ \max_{1 \leq i \leq m} \min_{1 \leq j \leq n} d(T_1^i, T_2^j), \max_{1 \leq j \leq n} \min_{1 \leq i \leq m} d(T_1^i, T_2^j) \right\} \quad (7)$$

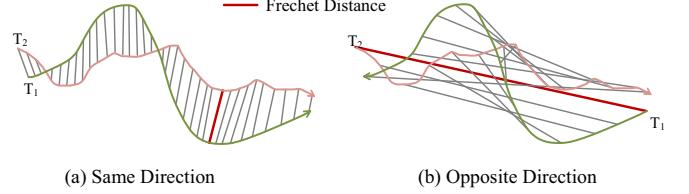


Fig. 5. Frechet Distance

In Figure 4(e), we have $\text{Hausdorff}(Q_1, Q_2) = 3.61$.

Frechet Distance. Frechet distance [2], [93] is similar to the Hausdorff distance. The main difference is that Hausdorff distance does not consider for the direction of two trajectories, while Frechet distance does as depicted in Figure 5. Maurice Fréchet [2] proposed this distance by taking an example of walking a dog, which has vividly explained the definition of it. Suppose that a man walks his dog on a leash. Though his dog may has different trajectory from him, they move in the same direction. And Frechet distance between two trajectories of man and dog is the shortest length of the leash that satisfies the characteristics (e.g., speed, traveling time) of two trajectories. Frechet distance is sensitive to noises with the time complexity of $O(mn)$. However, it is commonly used for trajectories with different lengths or sampling rates. Since Frechet distance is non-metric, many variants are proposed to make it be metric [15]. The most common one is discrete Frechet distance [72] with the time complexity of $O(mn)$. Thus, we use discrete Frechet distance in our experiments for its metricity. For simplicity, we use Frechet distance to denote discrete Frechet distance in the rest of this paper. Formally, discrete Frechet distacne

is defined as:

$$\text{Frechet}(T_1, T_2) = \begin{cases} \max_{1 \leq i \leq m} d(T_1^i, T_2^i) & \text{if } n = 1 \\ \max_{1 \leq i \leq n} d(T_1^i, T_2^i) & \text{if } m = 1 \\ \max \left\{ \begin{array}{l} d(T_1^m, T_2^n), \\ \min \left\{ \begin{array}{l} \text{Frechet}(T_1, \text{Head}(T_2)), \\ \text{Frechet}(\text{Head}(T_1), T_2), \\ \text{Frechet}(\text{Head}(T_1), \text{Head}(T_2)) \end{array} \right\} \end{array} \right. & \text{otherwise} \end{cases} \quad (8)$$

In Figure 4(f), we have $\text{Frechet}(Q_1, Q_2) = 4.12$.

Distributed settings. To accelerate the trajectory similarity computation, distributed techniques [57], [73], [82], [87] are developed, where DITA [57] and REPOSE [87] are two representative methods designed for point-based measures.

DITA. DITA [57] is a distributed in-memory trajectory analytics system, using several classical point-based measures such as DTW and Frechet. DITA chooses STR Partition algorithm [34] to partition trajectory points. It uses R-tree [26] as the local index, and designs a trie-like index as the global index. Accordingly, a trie-like partition algorithm is also used in the local index, and several pruning optimization methods are developed to improve the efficiency of similarity search and join in the distributed environment.

REPOSE. REPOSE [87] is a distributed in-memory system destined for trajectory similarity search, which supports multiple distance measures, including Hausdorff, Frechet, DTW, LCSS, EDR and ERP. Zheng et al [87] discretize trajectories into reference trajectories and construct a trie-like index (i.e., RP-Tree) on reference points of reference trajectories. It is worth mentioning that, REPOSE tends to divide similar trajectories into different partitions as much as possible to achieve load balancing.

3.1.2 Segment-based Measures.

Segment-based measures compute the similarity among trajectories, which are consisted of line segments. A line segment is a pair of points (p_1, p_2) , and its length is the Euclidean distance between p_1 and p_2 . Based on this, we introduce 5 segment-based measures (i.e., EDwP, LIP, OWD, Seg-Frechet and Seg-Hausdorff) and a distributed method (i.e., DFT) designed for segment-based measures.

Edit Distance with Projections (EDwP). EDwP distance [8], [51], [61] employs a parameter-free approach and adapts to non-uniform sampling rates through dynamic interpolation and projections. It transforms trajectory points into line segments, and uses insertion and replacement operations to compute the edit distance (see Figure 6). The time complexity of EDwP is $O(mn)$, and EDwP distance between two trajectories T_1 and T_2 is defined as:

$$\text{EDwP}(T_1, T_2) = \begin{cases} 0 & \text{if } m = n = 0 \\ \infty & \text{if } m = 0 \\ & \text{or } n = 0 \\ \min \left\{ \begin{array}{l} \text{EDwP}(T_1, \text{ins}(T_2, T_1)), \\ \text{EDwP}(\text{ins}(T_1, T_2), T_2), \\ \text{EDwP}(\text{Head}(T_1), \text{Head}(T_2)) + \\ (\text{rep}(T_1^1, T_2^1) \times \text{Coverage}(T_1^1, T_2^1)) \end{array} \right. & \text{otherwise} \end{cases} \quad (9)$$

where $\text{rep}(\cdot)$ and $\text{ins}(\cdot)$ denote the operations of replacement and insertion respectively. Specifically, $\text{rep}(e_1, e_2) = d(e_1.s, e_2.s) + d(e_1.e, e_2.e)$, where $e_i.s$ and $e_i.e$ represent the first and last point of line segment e_i . $\text{Coverage}(e_1, e_2) = \text{length}(e_1) + \text{length}(e_2)$ is used to measure the cost when replacing. In addition, $\text{ins}(e_1, e_2)$ effectively divides the line

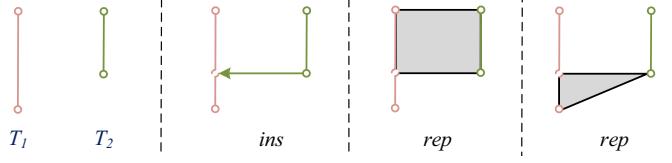


Fig. 6. EDwP Distance

segment e_1 into two segments by projecting e_2 onto it. In Figure 4(f), we have $\text{EDwP}(Q_1, Q_2) = 5.14$.

Locality In-between Polygons (LIP). LIP [3], [48] is a shape-based measure, which considers the areas of polygons formulated between intersection points of two trajectories. As shown in Figure 4(g), each polygons has different areas. LIP sets a weight (i.e., the perimeter of the polygon) for each area to adjust the effect of polygons. Nikos rt al. [48] extend LIP to a spatio-temporal (i.e., time-aware) distance called STLIP, and also propose a speed-pattern spatio-temporal distance called SPSTLIP. The time complexity of LIP, STLIP and SPSTLIP are all $O((m + n)\log(m + n))$. Formally, the LIP is defined as:

$$w_i = \frac{\text{Length}_{T_1}(I_i, I_{i+1}) + \text{Length}_{T_2}(I_i, I_{i+1})}{\text{Length}_{T_1} + \text{Length}_{T_2}} \quad (10)$$

$$\text{LIP}(T_1, T_2) = \sum_{\forall \text{Polygon}_i} \text{Area}_i \cdot w_i \quad (11)$$

In Figure 4(g), we have $\text{LIP}(Q_1, Q_2) = 4.04$.

Ont-way Distance (OWD). OWD [17], [21], [44] supports two representations of trajectories (i.e., linear representation and grid representation) as shown in Figure 4(h). Lin et al. [21] propose OWD based on linear representation, which transforms point-based trajectory data into segment-based. It measures the average minimal distance from each point in T_1 to the trajectory. Thus, OWD is an asymmetric measure according to Definition 4. Formally, the distance between a point and a trajectory $D(p, T)$, and OWD distance of linear representation are defined as below.

$$D(p, T) = \min_{q \in T} ED(p, q) \quad (12)$$

$$\text{OWD}(T_1 \rightarrow T_2) = \frac{1}{|T_1|} \left(\int_{p \in T_1} D(p, T_2) dp \right) \quad (13)$$

$$\text{OWD}(T_1, T_2) = \frac{1}{2} (\text{OWD}(T_1 \rightarrow T_2) + \text{OWD}(T_2 \rightarrow T_1)) \quad (14)$$

Considering the high computation cost of linear representation (i.e., the time complexity of $O(mn)$), OWD is extended to grid representation called OWD^g , where trajectory points are mapped into grid cells according to their spatial information. Thus, OWD only computes the distances between grid cells (instead of sample points) and grid-based trajectories. The time complexity is reduced to $O(MN)$, where M and N denote the number of grid cells occupied by two trajectories, respectively. Compared to OWD, OWD^g is more popular due to low time complexity [59]. Formally, OWD^g between two grid-based trajectory T_1^g and T_2^g is defined as:

$$D^g(g, T^g) = \min_{g' \in T^g} \text{distance}(g, g') \quad (15)$$

$$\text{OWD}^g(T_1^g \rightarrow T_2^g) = \frac{1}{|T_1^g|} \sum_{p \in T_1^g} D^g(p, T_2^g) \quad (16)$$

$$\text{OWD}^g(T_1^g, T_2^g) = \frac{1}{2} (\text{OWD}^g(T_1^g \rightarrow T_2^g) + \text{OWD}^g(T_2^g \rightarrow T_1^g)) \quad (17)$$

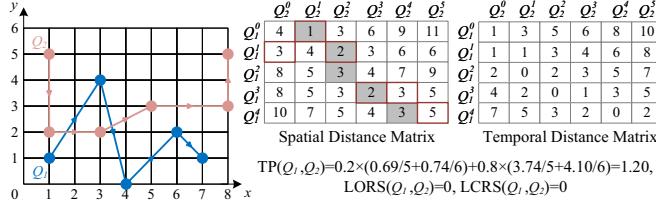


Fig. 7. Non-learning Based Measures in Road Network

Where $\text{distance}(\cdot)$ computes the distance of two grid cells. As depicted in Figure 4(h), we have $OWD(Q_1, Q_2)=1.67$ and $OWD^g(Q_1, Q_2)=0.25$.

Note that, when dealing with sparse data with a large spatial span, the space overhead of OWD^g is quite large.

Seg-Frechet and Seg-Hausdorff. Inspired by Frechet and Hausdorff distance, Seg-Frechet and Seg-Hausdorff distance [73] are proposed for segment-based trajectories, by changing points to line segments. Segment-based trajectories use the sequence of line segments to capture the path information. For example, in Figure 2, we have Seg-Frechet(Q_1, Q_2)=3.61.

Distributed setting. Xie et al. [73] develop a distributed framework (i.e., DFT) based on Spark that supports Seg-Frechet and Seg-Hausdorff measures. DFT uses STR partition algorithm [34] to partition segments, while constructs a dual R-tree as the global index and a general R-tree as the local index. DFT is the first distributed method to support fast trajectory similarity computation.

3.2 Trajectory Similarity in Road Network

In this subsection, we review 7 distance measures designed for road network constrained trajectories in terms of point-based and segment-based measures. Different from free space, trajectories in road network are usually denoted by road vertices or road segments. Here, we consider the running example in road network, where the vertices and edges of grids denote the road vertices and segments, respectively. Figure 7 depicts Q_1 and Q_2 , with the corresponding spatial and temporal distance matrix to store the road network distances between vertices of two trajectories.

3.2.1 Point-based Measures.

Point-based measures in road network are utilized to compute the similarity among trajectories that are consisted of road vertices. We mainly introduce 5 point-based measures (i.e., NetERP, NetEDR, NetDTW, NetLCSS and TP) below.

NetERP, NetEDR, NetDTW and NetLCSS. NetERP [31], NetEDR [31], NetDTW [74] and NetLCSS [19] are expanded from classic measures in free space. They first map original trajectories into road network paths that consist of vertices or segments. Then, they define similarity measures based on classic distance measures such as ERP, EDR, DTW and LCSS, generally by aggregating the distances between road vertices or segments of two trajectories. Note that, these measures employ the shortest path distance (instead of Euclidean distance) between two road vertices in the graph. Thus, NetERP, NetEDR, NetDTW and NetLCSS have the same features as corresponding classic measures, and can be used in road network. For the example in Figure 7, we use $|x_1 - x_2| + |y_1 - y_2|$ denoted

as the distance between two road vertices $(x_1, y_1), (x_2, y_2)$. Then we have $\text{NetERP}(Q_1, Q_2)=20.00$, $\text{NetEDR}(Q_1, Q_2)=4$, $\text{NetDTW}(Q_1, Q_2)=19.00$, and $\text{NetLCSS}(Q_1, Q_2)=1$.

TP. TP [56] is a measure that considers both the spatial and temporal similarities. Shang et al. [56] assume that the point is 3-dimension, i.e., $v = (p, t) = ((\text{latitude}, \text{longitude}), \text{timestamp})$. The spatial and temporal similarities (i.e., Sim_S and Sim_T) are:

$$D'(v.p, T) = \min_{q \in T} d'(v.p, q.p) \quad (18)$$

$$D'(v.t, T) = \min_{q \in T} |v.t - q.t| \quad (19)$$

$$\text{Sim}_S(T_1, T_2) = \frac{\sum_{T_1^i \in T_1} e^{-D'(T_1^i.p, T_2)}}{|T_1|} + \frac{\sum_{T_2^j \in T_2} e^{-D'(T_2^j.p, T_1)}}{|T_2|} \quad (20)$$

$$\text{Sim}_T(T_1, T_2) = \frac{\sum_{T_1^i \in T_1} e^{-D'(T_1^i.t, T_2)}}{|T_1|} + \frac{\sum_{T_2^j \in T_2} e^{-D'(T_2^j.t, T_1)}}{|T_2|} \quad (21)$$

Here, $d'(\cdot, \cdot)$ denotes the shortest path distance between two road vertices in the graph. The spatial and temporal similarities are combined linearly to obtain the spatio-temporal similarity, i.e., $\text{Sim}_{ST}(T_1, T_2) = \lambda \cdot \text{Sim}_S(T_1, T_2) + (1 - \lambda) \cdot \text{Sim}_T(T_1, T_2)$, where λ ($0 \leq \lambda \leq 1$) is a parameter to adjust the relative importance of spatial and temporal similarities. Considering the temporal information, we randomly add timestamps for Q_1 and Q_2 in Figure 7, i.e., $Q_1 = (1, 1, 1), (3, 4, 3), (4, 0, 4), (6, 2, 6), (7, 1, 9)$, $Q_2 = (1, 5, 2), (1, 2, 4), (3, 2, 6), (5, 3, 7), (8, 3, 9), (8, 5, 11)$. when $\lambda=0.2$, we have $\text{TP}(Q_1, Q_2)=1.20$.

3.2.2 Segment-based Measures.

Segment-based measures in road network measure the similarity among trajectories consisted of road segments. We mainly introduce 2 segment-based measures (i.e., LORS and LCRS) and a distributed framework (i.e., DISON).

Longest Overlapping Road Segments (LORS). LORS [69] is a distance measure based on the length of overlapped edges (i.e., road segments) with the time complexity of $O(mn)$. Given two trajectories T_1 and T_2 , where the i -th ($1 \leq i \leq m$) segment of T_1 is denoted as T_1^i , LORS measure is defined formally as:

$$\text{LORS}(T_1, T_2) = \begin{cases} 0 & \text{if } m = 0 \text{ or } n = 0 \\ \text{length}(T_1^m) & \text{if } T_1^m = T_2^n \\ + \text{LORS}(\text{Head}(T_1), \text{Head}(T_2)) & \text{if } T_1^m \neq T_2^n \\ \max \left\{ \text{LORS}(T_1, \text{Head}(T_2)), \text{LORS}(\text{Head}(T_1), T_2) \right\}, & \text{otherwise} \end{cases} \quad (22)$$

where $\text{length}(\cdot)$ denotes the length of the edge. According to Eq. 22, LORS does not need to compute the shortest path distance in the graph, which is relatively efficient. In Figure 7, we have $\text{LORS}(Q_1, Q_2)=0$ (Q_1 and Q_2 have no overlapping segments).

Longest Common Road Segments (LCRS). Similar to LCSS, LCRS [82] finds the longest common road segments between two trajectories, and can be used in several applications such as carpooling. The time complexity of LCRS is also $O(mn)$, and LCRS satisfies symmetry. It is formally defined as:

$$\text{LCRS}(T_1, T_2) = \frac{\text{LORS}(T_1, T_2)}{m + n - \text{LORS}(T_1, T_2)} \quad (23)$$

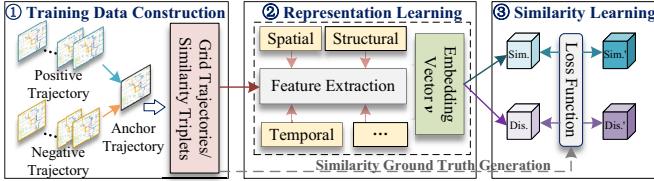


Fig. 8. The Processing Pipeline of Learning-based Models

For the example in Figure 7, $\text{LCRS}(Q_1, Q_2)=0.00$ (Q_1 and Q_2 have no common segments).

Distributed setting. Yuan et al. [82] extend standalone LCRS to a distributed framework called DISON, which supports road-network constraint similarity search and join. DISON first uses STR partition algorithm to partition the road segments, and then it builds a two-layer index with hashmap as the global index and an inverted index as the local index. Note that, DISON is unique distributed framework for similarity measure in road network.

4 LEARNING BASED MEASURES

As mentioned in Section 3, non-learning measures generally have a time complexity of $O(mn)$, which is high. To address this, learning-based methods are developed to reconstruct the input data of high-dimension into a new representation of low-dimension [37], [43], [80].

t2vec [37] designs a trajectory deep learning framework to achieve robustness to low sample rates and noises. It maps a trajectory into a d -dimensional embedding vector. Note that, t2vec is not a similarity measure, but a representation method to transform trajectories into vectors.

Inspired by t2vec, many studies employ different deep learning frameworks in free space and road network, to learn approximate distance functions for non-learning measures. Generally, as shown in Figure 8, they first select similar and dissimilar trajectories (i.e., positive and negative trajectories) as anchor trajectories to obtain grid-based trajectories [78] or similarity triplets [19], [27], [86]. Then, features (in terms of spatial, temporal, structural and so on) are extracted, while trajectory embedding vectors are generated using deep representation learning. Finally, the learning (dashed rectangle in Figure 8) can stop until the trajectory similarities and dissimilarities evaluated on the embedding vectors are close to the ground-truth. Here, results calculated using non-learning measures are taken as ground-truth. Note that, the time complexity of similarity learning among vectors is linear. In view of this, we review 4 representative learning-based measures.

NEUTRAJ. The first learning-based trajectory similarity measure is NEUTRAJ [78], which is based on neural metric learning. Specifically, NEUTRAJ first maps trajectories into grid-trajectories. Then, it samples trajectories as seeds, and uses their pair-wise similarities and dissimilarities as guidance. Finally, NEUTRAJ uses Long Short-Term Memory (LSTM) model to generate the embedding vectors, and approximates various non-learning similarity computations (as depicted in Figure 8) with the complexity of $O(m + n)$. Note that, embedding vectors generated by NEUTRAJ can preserve the spatial information of trajectories. In addition, in order to improve the performance of deep representation learning methods on long trajectories, TrajGAT [79]

TABLE 2
Evaluation Benchmark

Dimension	Varying Cases	Adjustable Parameters
Effectiveness	vs. <i>length</i>	L (%): 20, 60, 100
	vs. <i>shape</i>	Four typical geometrics
Robustness	vs. <i>sampling</i>	S (%): 10, 20, 40
	vs. <i>noise</i>	N (%): 13, 16, 19/10, 20, 30
Efficiency	Top- k query	/
	vs. <i>length</i>	L (%): 20, 60, 100
Scalability	vs. <i>cardinality</i>	O_r (%): 20, 60, 100

is proposed, which is based on graph attention networks (GATs), Transformer and a quad-tree index for effectively embedding trajectories.

Traj2SimVec. Although NEUTRAJ has greatly reduced the time complexity, it needs a pre-training process to compute the similarity among all seed trajectories, which incurs a quadratic time complexity during training. Thus, Traj2SimVec [86] is proposed to improve the training efficiency by simplifying training trajectories into triplet training samples. And the training time complexity is reduced to $O(\log n)$, where n denotes the average length of training trajectories. Similar to Traj2SimVec, T3S [77] and TMN [76] are proposed based on attention networks, where the former considers dissimilar trajectories while the latter focuses on the inter-information between trajectories.

GTS. Different from the above two methods that are designed in free space, a Graph Neural Network (GNN) based approach named GTS [27] is designed to measure the trajectory similarity in road network. In order to reflect the information on the road network, GTS learns the representation of point-of-interest (POI) in spatial network and the relationship between trajectories, between POIs, or between trajectories and POIs.

ST2Vec. ST2Vec [19] is also proposed for spatio-temporal trajectory similarity computation in road network. Considering the complex temporal information in trajectory data [59], ST2Vec with a time complexity of $O(d)$ captures both spatial and temporal features, and fusions these features to obtain spatio-temporal embedding vectors based on GNN and LSTM. Also, ST2Vec applies training triplets samples to reduce the training cost, and sets spatial and temporal weights that can be adjusted for flexible analysis.

Note that, due to the similarity between different learning-based methods, we mainly evaluate NEUTRAJ [78], Traj2SimVec [86], GTS [27], ST2Vec [19] in Section 6 for simplicity.

5 EVALUATION BENCHMARK

Motivation. Since the evaluations on such a big family of trajectory similarity measures have never been organized, we are inspired to propose the first standard evaluation framework to examine the function of each measure. The primary goals of a distance measure are being effective, robust, efficient, and scalable. These four desiderata are often in tension with each other. In view of these, the evaluation framework must enable studying the capability of each measure in terms of multi-aspects. On the other hand, trajectory data usually have various characteristics such as variable lengths, complex shapes (especially for vehicle trajectories), varying samplings, outlier noises, etc.

Specifically, the variable lengths and complex shapes are intrinsic properties [89], while the varying samplings and outlier noises are usually caused by other factors [59] such as battery power and urban building effects. Overall, we build an evaluation framework to benchmark the trajectory measures under as many real-life settings as possible, objectively studying the capability of each measure in terms of four performance aspects. Table 2 summarizes all the evaluation dimensions and corresponding scenarios. According to Table 2, we evaluate each trajectory similarity measure from four dimensions as below.

(i) *Effectiveness vs. length and shape*. Length and shape are considered as intrinsic properties of trajectory data [36], which requires an effective measure. However, as the similarity of two trajectories does not have a ground truth [33], we cannot determine whether the value computed by trajectory measures is correct or not. It is also meaningless to cross-compare the distance values returned by different measures. To this end, we adopt a qualitative way to visualize the Top- k similarity query results returned by different trajectory measures, for the same query trajectory. Thus, we give an intuitive analysis when the query trajectory is selected with different lengths or shapes to show the effectiveness of each measure.

Specifically, we use parameter L to change the trajectory length. For example, $L = 20$ means to select the first 20% points from the original trajectory as a deformed trajectory. Such length-oriented evaluation enables us to see how each distance measure performs when the length of the trajectory changes from short to long, which is useful in an online setting that involves trajectory evolutions. In terms of varying shapes, we select four typical geometrics, i.e., straight line, polyline without overlaps, polyline with overlaps, and round line, to explore how each similarity measure performs when the query trajectories are of different shapes.

(ii) *Robustness vs. sampling and noise*. Due to objective factors, quality issues commonly exist during data collection. For example, a taxi driver may alter the default device sampling rate to reduce the power consumption [88]. In that case, even for the same trajectory, this may result in sparse and dense segments. In addition, noise may occur in GPS points, which is usually caused by low satellite visibility or urban canyons that affect signal quality. Consequently, given a query trajectory, a robust trajectory similarity measure is expected to retrieve the same query results, even being faced with samplings or noise effects on the queried datasets.

Specifically, we use parameter S to denote the percentage of sampling points in each trajectory. For example, $S = 20$ means that we sample 20% points from a trajectory as a deformed trajectory. Note that, sampling transformation is meaningful and useful for free space trajectories, while meaningless and useless for road network constrained trajectories after map matching. We use parameter N to add outlier points. For example, $N = 10$ denotes to sample 10% points from a trajectory, and then add an outlier for each point. Here, we utilize Gaussian noise [5] to add an outlier p' for each filtered point p , i.e., $p' = p + \Delta \cdot p \sim Gaussian(0, 1)$, where Δ denotes the radius range of outliers.

(iii) *Efficiency vs. similarity computation*. An efficient trajectory similarity measure is expected to efficiently compute

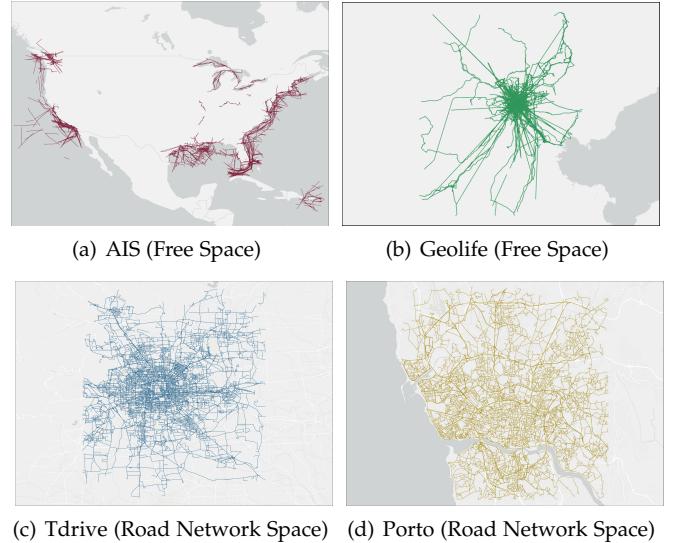


Fig. 9. Visualization of The Datasets Used

the distance value between any pair of trajectories, which especially is important for online trajectory applications. Thus, we directly use the similarity computation as the evaluation task.

(iv) *Scalability vs. length and cardinality*. Despite how the average trajectory length or the data cardinality (i.e., the number of moving objects) of the trajectory dataset changes, a scalable measure should show stable inter-trajectory distance computation performance. Specifically, we proceed to use $L(\%)$ to control the average length of trajectories in the dataset. In addition, we use another parameter O_r to control the percentage of moving objects w.r.t. all the objects, i.e., the trajectory dataset to be queried/processed.

6 EXPERIMENTAL EVALUATION

We first present experimental settings. Then, based on the evaluation benchmark proposed in Section 5, we study the capability of all trajectory measures from four major aspects, where the insights and discussions of the objective experiments are also provided. Furthermore, we conduct a case study about the metric measures which can be used for indexing and pruning.

6.1 Experimental Settings

Datasets. We utilize four real-world trajectory datasets for experimental evaluations, including AIS [49], Geolife [91], T-Drive [50] and Porto [28], all of which are widely used in previous trajectory similarity analyses. The visualization of the four datasets is shown in Figure 9, from which we can find that AIS has a larger spatial span and a very uneven data distribution compared to Geolife.

- AIS is a ship dataset collected in free space, that contains 48 GB trajectory points of vessels during 12 months, from Jan. 1 to Dec. 1, 2019. The points are collected periodically with sampling rates changing between 1 and 90 seconds.
- Geolife is a dataset which contains about 2500w GPS trajectories collected from 182 users in Beijing, from April, 2007 to August, 2012. And the sampling rates

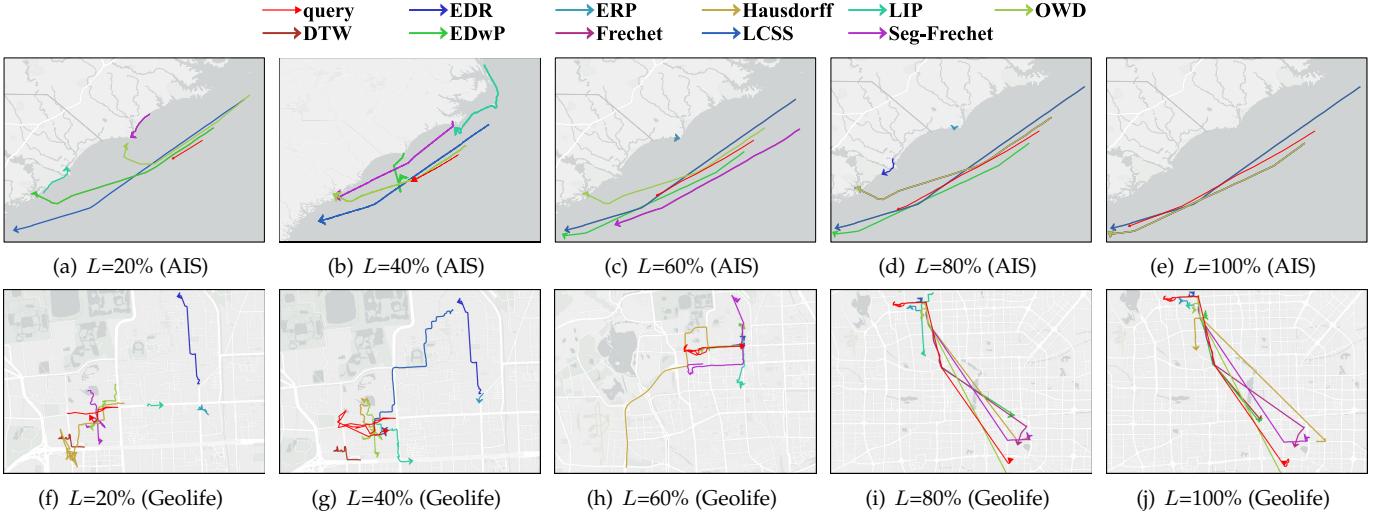


Fig. 10. Effectiveness Study of Non-learning based Measures vs. Trajectory Length

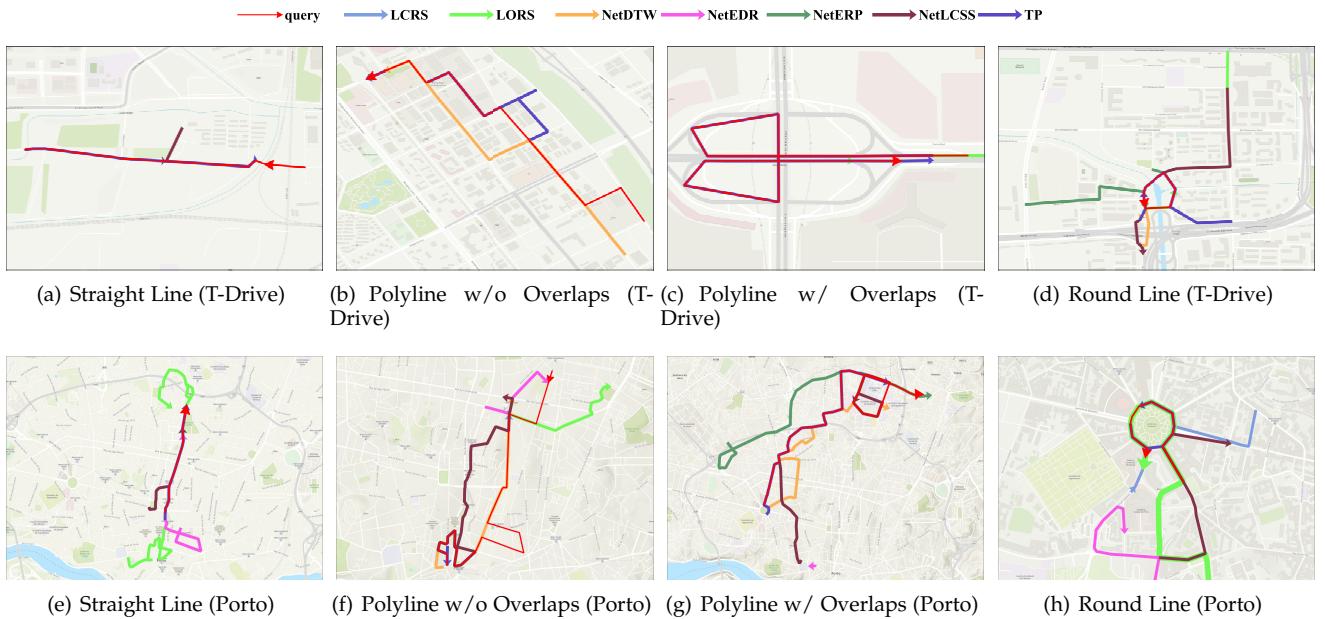


Fig. 11. Effectiveness Study of Non-learning based Measures vs. Shape

are about 2s. Geolife records the different transportation modes of users, so that the trajectories are with speed mutation.

- T-Drive is a trajectory dataset generated by taxis in Beijing road networks during 7 days, from Feb. 2 to Feb. 8, 2008. In this dataset, there are about 1.5 million sample points in total. The GPS points are collected with sampling rates varying between 1 and 177 seconds.
- Porto is a dataset describing trajectories performed by all the 442 taxis running in the city of Porto, in Portugal. There are about 170w trajectories in Porto collected with sampling rates about 15s.

In the experiments, we adopt AIS and Geolife for free space oriented trajectory measure evaluations and use T-drive and Porto for road network constrained trajectory measure evaluations. Specifically, we remove trajectories that contain less than 5 sampling points, and map match [75] all the trajectories in T-Drive and Porto to the corresponding road network from OpenStreetMap [47]. After such prepro-

cessing, we obtain 50, 000 trajectories in each dataset. The average lengths of AIS, Geolife, T-Drive and Porto are about 500, 350, 150 and 60, respectively. Considering the high complexity of similarity computation and the capability limitation of a single machine, we randomly select 10, 000 trajectories from all datasets for evaluations in the standalone processing mode, while we use all the trajectories from the four datasets for evaluations in the distributed settings.

Compared Measures. As shown in Table 1, we evaluate 25 representative trajectory similarity measures/methods, including:

(i) *Non-learning based measures* (21). We first study the standalone-based 17 measures including free space or road network oriented ones, i.e., DTW, LCSS, EDR, EDwP, ERP, Hausdorff, Frechet, LIP, OWD, Seg-Frechet, LORS, TP, NetERP, NetEDR, NetDTW, NetLCSS, and LCRS, for effectiveness/robustness/efficiency/scalability evaluations. Then, we study the distributed-based 4 methods including free space or road network oriented ones, i.e., DFT, DITA,

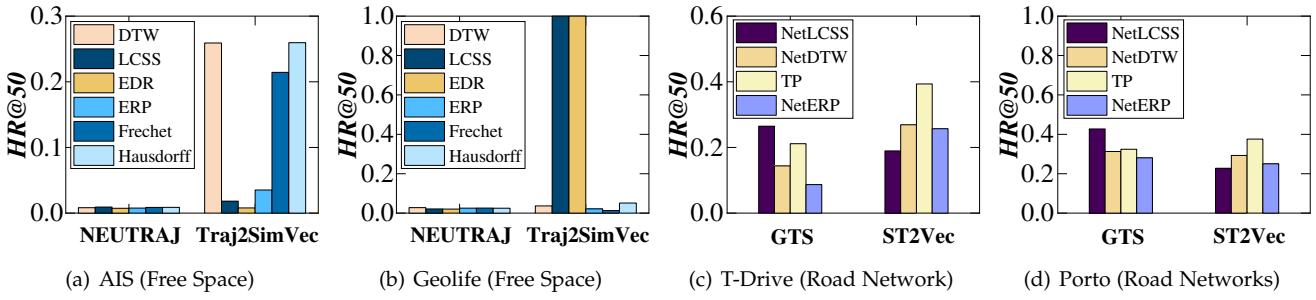


Fig. 12. Effectiveness of Learning-based Measures

REPOSE, and DISON, only for efficiency and scalability evaluations, as these distributed implementations do not modify the measures but leverage distributed techniques for boosting. It is worth mentioning that, in order to objectively compare the differences of each measure, we do not use any indexing structure in the evaluation.

(ii) *Learning-based measures* (4). In this category, all existing measures, i.e., NEUTRAJ [78], Traj2SimVec [86], GTS [27], and ST2Vec [19], are standalone-based. Thus, we evaluate their effectiveness/robustness/efficiency/scalability in free-space or road network contexts. Note that, the target of learning-based models is to approximate the inter-trajectory similarity relations computed by using non-learning measures and all existing learning-based measures can only support point-based similarity learning. Therefore, for each model, we evaluate its capability of approximating point-based trajectory similarity measures, including DTW, LCSS, EDR, EDwP, ERP, Hausdorff, and Frechet in free space, as well as NetDTW, NetLCSS, NetERP, and TP in road network.

Hyperparameters. For similarity measures EDR and LCSS used in free space, we set the similarity threshold used in similarity computation as 30km. For similarity measures NetEDR and NetLCSS used in road network, we set the similarity threshold to 1km. Besides, we set the gap point of ERP and NetERP as the centroids of datasets. To vary noises in benchmark settings, we set $\Delta lat = \Delta lon = 0.1$ for AIS, $\Delta lat = \Delta lon = 0.0005$ for Geolife, and $\Delta lat = 0.008$, $\Delta lon = 0.007$ for T-Drive and Porto.

For standalone-based measures, we conduct experiments on an 8-core Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz processor. For distributed-based measures, we conduct experiments on a cluster consisting of 9 nodes with nine 10-core Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz processors. Each node ran CentOS7 system with Hadoop 2.6.5 and Spark 2.2.0. For learning-based measures, we set the hyperparameters based on the performance of each model. Specifically, for NEUTRAJ [78], the batch size is set to 60 and the initial learning rate is 0.01. For Traj2SimVec [86], the batch size is set to 64 and the embedding size of GCN is 128. For GTS [27] and ST2Vec [19], the batch size is set to 128 and the initial learning rate is 0.1. The hidden size of all LSTMs used in these measures is set to 128. Moreover, we split each dataset into train set, validate set, and test set in the ratio of 3:1:6. All the experiments of learning-based methods are conducted on a server with GeForce RTX 3090, 2.40GHz GPU, and 24GB RAM.

6.2 Effectiveness Evaluation

6.2.1 Non-learning based measures.

Specifically, for these handcrafted designed classic distance measures, we utilize a visualization way to plot the retrieved Top- k results for the same query trajectory when using different distance measures.

Effectiveness vs. Length. Figures 6(a)–6(j) plot the Top-1 query results for the same query trajectory (denoted by red line) under different query lengths, i.e., varying L from 20% to 100%. Here, we only evaluate the AIS, Geolife datasets, and free-space oriented measures, as the free-space based trajectories usually have longer lengths than vehicle trajectories in road networks.

According to Figure 6, we yield the following insights:

- (i) When $L = 100\%$, most measures return similar results. Specifically, in AIS, LCSS and Hausdorff share the same result, while DTW, Frechet, OWD, and Seg-Frechet share another same result. And in Geolife, DTW, EDwP, OWD, Hausdorff, Frechet and Seg-Frechet share the similar results with the same transportation mode as query trajectory, where only Hausdorff returns result with opposite direction of the query.
- (ii) When the length of query trajectory varies, in AIS, DTW, EDwP, Hausdorff, and OWD always share the same results, which implies these measures are suitable for evolution or streaming analysis. In Geolife, DTW, Hausdorff, EDwP, Frechet, Seg-Frechet, and OWD always find results with same transportation modes as query trajectory.
- (iii) When the query length is shorter in AIS, i.e., in Figures 3(a)–3(c), EDR performs relatively worse than others, as its returned results are too far away from the query trajectory on the map.
- (iv) LIP has the most quite different query results under different query lengths, as it is affected by trajectory shapes, making LIP not robust to different query lengths.
- (v) The results of various measures returned in AIS and Geolife are a little different. This is because the trajectories in AIS have a larger spatial span between two points, which make various measures easy to find much more similar trajectories with query. And in Geolife, there are quite different features in one trajectory due to the transportation mode transformation. Thus, when processing such trajectories, most measures are sensitively to be affected by different lengths.

Effectiveness vs. Shape. Figures 7(a)–7(h) plot the Top-1 query results for four different query trajectories (denoted by red line), each of which owns a typical spatial shape, i.e., straight, polyline without overlaps, polyline with overlaps, and round. Here, we only evaluate the T-drive and Porto datasets, and road-network oriented trajectory measures, as

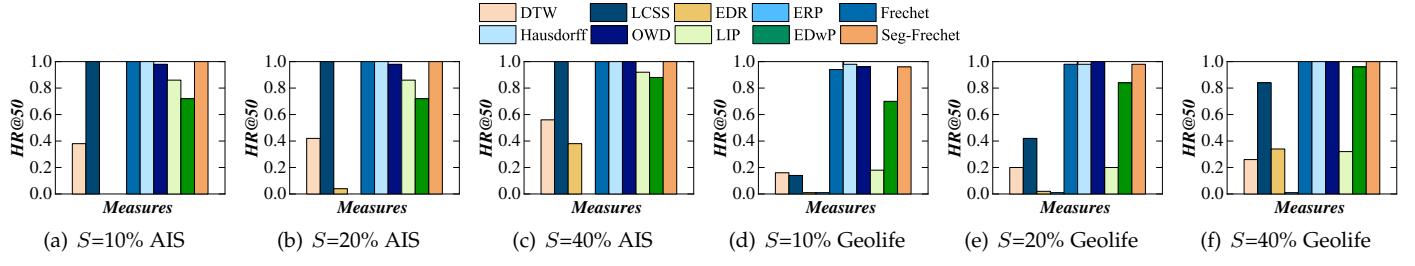


Fig. 13. Robustness of Non-learning Measures vs. Sampling

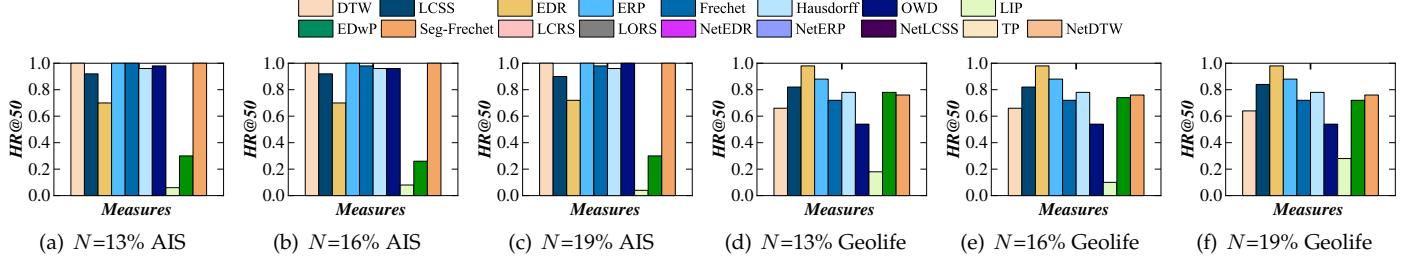


Fig. 14. Robustness of Non-learning based Measures vs. Noise

TABLE 3
Robustness of Learning-based Measures vs. Sampling Rate (AIS)

Sample Rate (%)	Measures	DTW	LCSS	EDR	ERP	Frechet	Hausdorff
$S=10$	NEUTRAJ	0.0087	0.0075	0.0073	0.0082	0.0086	0.0087
	Traj2SimVec	0.2515	0.0016	0.0078	0.0356	0.2067	0.2520
$S=20$	NEUTRAJ	0.0087	0.0083	0.0087	0.0080	0.0091	0.0091
	Traj2SimVec	0.2603	0.0019	0.0077	0.0352	0.2151	0.2610
$S=40$	NEUTRAJ	0.0090	0.0080	0.0082	0.0084	0.0086	0.0095
	Traj2SimVec	0.2590	0.00183	0.0079	0.0351	0.2151	0.2602
$S=100$	NEUTRAJ	0.0082	0.0094	0.0074	0.0076	0.0087	0.0088
	Traj2SimVec	0.25905	0.0018233	0.0079	0.0352	0.2143	0.2705

TABLE 4
Robustness of Learning-based Measures vs. Sampling Rate (Geolife)

Sample Rate (%)	Measures	DTW	LCSS	EDR	ERP	Frechet	Hausdorff
$S=10$	NEUTRAJ	0.0281	0.0217	0.0140	0.0156	0.0178	0.0166
	Traj2SimVec	0.0106	0.3312	0.3312	0.0271	0.0158	0.0110
$S=20$	NEUTRAJ	0.0277	0.0204	0.0113	0.0179	0.0200	0.0183
	Traj2SimVec	0.0094	1.000	1.000	0.0245	0.0158	0.0110
$S=40$	NEUTRAJ	0.0277	0.0206	0.0121	0.0210	0.0220	0.0214
	Traj2SimVec	0.0094	1.000	1.000	0.0245	0.0158	0.0112
$S=100$	NEUTRAJ	0.0277	0.0212	0.0204	0.0252	0.0258	0.0249
	Traj2SimVec	0.0366	1.000	1.000	0.0215	0.0131	0.0513

the vehicle trajectories usually have much more complex geometrics due to the constraints of road networks.

According to Figure 7, we yield the following insights: (i) NetEDR and NetLCSS perform the worst in all cases. For example, in Figure 7(a), for a straight trajectory, both NetEDR and NetLCSS return back-and-forth trajectories, while all of the remaining measures return straight trajectories with close results. (ii) For query trajectory without/with overlaps cases, all of other measures return similar results. (iii) For the most complex case where the query trajectory features round shapes in Figure 7(d) and 7(h), only LORS, LCRS, and TP return trajectories with round information, where LORS performs the best. This is because LORS finds overlapping road segments, making it capable to match the round shape in the query trajectory.

6.2.2 Learning based measures

For learning-based methods, since they target approximating non-learning based measures, their effectiveness de-

pends on the capability of approximating the non-learning based measures. Specifically, for a query trajectory, we set its Top- k query results using non-learning based distances (e.g., DTW) as the ground truth. Then, we use a metric Top- k hitting ratio (i.e., $HR@k$) to evaluate the effectiveness of learning-based methods. In other words, $HR@k$ describes the degree of overlap between the Top- k similarity query results returned by the learning-based models and the ground truth results. For simplicity, we set $k=50$ in the remaining experimental evaluation.

Figure 12 show the effectiveness of deep models for free space and road network oriented similarity learning, Based on which, we have the following insights: (i) In free space settings, although Traj2SimVec shows better approximation ability than NEUTRAJ, it performs very unstable across different similarity measures (e.g., LCSS, EDR, and ERP) in different datasets (e.g., Frechet and Hausdorff). This is because, these similarity measures use string-based

distance, whose information cannot be preserved in embeddings. And in Geolife, the uneven and complex features of trajectories make the approximation of various measures unstable. (ii) In road network settings, both GTS and ST2Vec perform stably, where ST2Vec performs better than GTS. This is because, ST2Vec leverages more sufficient spatial information (i.e., road networks) into similarity learning than POI information used in GTS. (iii) Overall, the learning-based models perform much better in road network than in free space, which implies the potential capability of learning techniques for road network constrained trajectory analyses.

6.3 Robustness Evaluation

According to Table 2, we proceed to study the robustness of each trajectory similarity measure under different samplings and noises.

6.3.1 Transformation procedures.

For non-learning based measures, given a specific similarity measure (e.g., DTW) and a randomly selected query trajectory T from the dataset: i) We first compute its Top- k similar trajectories via this similarity measure (e.g., DTW) and set that as the query ground-truth. ii) Then, we perform corresponding down-sampling or add-noise operations on the original dataset via adjustable parameters S and N in Table 2, resulting in several versions of transformed trajectory datasets. iii) We proceed to perform Top- k similarity query for the same query trajectory T in each transformed dataset to obtain corresponding Top- k results. A robust trajectory similarity measure should return close results from i) and iii), which can be also evaluated by $HR@k$ metric. For learning-based methods, we perform a similar procedure as non-learning based measures. The main difference is we compute Top- k similarity results in non-learning settings while we learn Top- k similarity results in learning settings.

6.3.2 Non-learning based measures.

We first evaluate the robustness of each non-learning measure to samplings and noises.

Effects of Sampling. We perform downsampling with $S(\%)$ of 10, 20, and 40. The original dataset corresponds to $S = 100$. Here, we only use the AIS and Geolife datasets, and free space oriented measures for evaluations, as the variation of sampling has little effect on map-matched trajectories in road networks.

The results are shown in Figure 13, from which we yield the following insights: (i) DTW, EDR, and ERP are very sensitive to sampling rates, whose performance increases as the sampling rate increases. This is in contrast to previous experience, as previous studies mainly use a small dataset with small spatial coverage. However, AIS has a large spatial span and Geolife has an uneven data distribution, in which case, similarity measures (i.e., DTW and ERP) that perform well in small urban areas [59] are highly affected by sampling rates. (ii) For AIS dataset, LIP and EDwP are slight sensitivity to sampling rates, as their $HR@50$ varies between 0.6 and 0.9. While in Geolife, LIP is sensitive to sampling rate. This is because the shape constantly varies during the transportation modes changes. And downsampling greatly affects the original shapes. (iii)

Frechet, Hausdorff, OWD, and Seg-Frechet are robust, as their $HR@50$ is easily beyond 0.9 even facing a very low sampling rate $S = 10\%$. This is because these measures rely on matched pairs, which performs stable when the number of sampling points changes.

Effects of Noise. Considering the data distributions of AIS and T-drive, we add noises with $N(\%)$ of 13, 16, and 19 for AIS and Geolife, and 10, 20, and 30 for T-drive and Porto. The original dataset corresponds to $N = 0$.

Figure 14 show the results, from which we have the following insights: (i) In free space settings, most of measures including DTW, LCSS, ERP, Frechet, Hausdorff, OWD, and Seg-Frechet show stable and high performance. This is because, measures such as LCSS and EDR use string distance to measure the inter-trajectory similarity, making them robust to noises. Also, as the trajectories in AIS are gathering locally and dispersing globally, there is no impact on grid-trajectory mapping of OWD when adding noises. While in Geolife, noises may slightly affect the effect of OWD. Moreover, LIP performs worse, which is in contrast to previous experience. This is because, LIP computes similarity according to the areas of polygons between point intersections. However, the areas of polygons sensitive to noise points. Also, EDwP performs differently on twp free space datasets. Because EDwP computes coverage which is sensitive to noises in AIS. (ii) In road network settings, all similarity measures show an obvious performance decrease when adding more noises. As there are many paths between two locations in road networks, changing some road segments of a trajectory may result in another new trajectory (i.e., another path), making most of the measures sensitive to noises. Besides, when extending free space oriented measures into the road network context, some noise-tolerant measures such as ERP and EDR become noise-sensitive and others such as DTW and LCSS still keep noise-tolerant, indicating that not all similarity measures are suitable for extension to the road networks.

6.3.3 Learning-based measures.

We proceed to evaluate the robustness of each learning-based measure to samplings and noises.

Effects of Sampling. We perform downsampling operations with the same $S(\%)$ settings of 10, 20, and 40 to obtain a series of transformed datasets. Then, we directly use the well-trained deep models on the original dataset to conduct the Top- k (i.e., $k = 50$) similarity queries on the transformed datasets to show the $HR@50$ performance. Table 3 and Table 4 shows the results. Based on that, we observe that all similarity functions learned by NEUTRAJ and Traj2SimVec are stable when varying sampling rates, which proves the high robustness of AI-based similarity learning to various sampling rates. This is because the embedding is not affected by sampling points.

Effects of Noise. We also add the same noises to that of non-learning settings to obtain a series of transformed datasets. Then, we directly use the well-trained deep models on the original dataset (i.e., $N = 0\%$) to conduct the Top- k ($k = 50$) similarity queries on the transformed datasets to show the $HR@50$ performance. Table 5, 6, 7 and 8 show the results on AIS, Geolife, T-drive and Porto datasets,

TABLE 5
Robustness of Learning-based Measures vs. Noise (AIS)

Noise (%)	Measures	DTW	LCSS	EDR	ERP	Frechet	Hausdorff
$N=0$	NEUTRAJ	0.0082	0.0094	0.0074	0.0076	0.0087	0.0088
	Traj2SimVec	0.25905	0.0018233	0.0079	0.0352	0.2143	0.2705
$N=13$	NEUTRAJ	0.0091	0.0080	0.0073	0.0079	0.0087	0.0085
	Traj2SimVec	0.2460	0.0017	0.0079	0.0345	0.2091	0.2583
$N=16$	NEUTRAJ	0.0092	0.0071	0.0080	0.0082	0.0091	0.0085
	Traj2SimVec	0.2464	0.0021	0.0080	0.0348	0.2053	0.2558
$N=19$	NEUTRAJ	0.0083	0.0071	0.0066	0.0081	0.0087	0.0084
	Traj2SimVec	0.2403	0.0017	0.0078	0.0334	0.2034	0.2540

TABLE 6
Robustness of Learning-based Measures vs. Noise (Geolife)

Noise (%)	Measures	DTW	LCSS	EDR	ERP	Frechet	Hausdorff
$N=0$	NEUTRAJ	0.0277	0.0212	0.0204	0.0252	0.0258	0.0249
	Traj2SimVec	0.0366	1.000	1.000	0.0215	0.0131	0.0513
$N=13$	NEUTRAJ	0.0276	0.0223	0.0203	0.0243	0.0247	0.0227
	Traj2SimVec	0.0094	1.000	1.000	0.0255	0.0158	0.0110
$N=16$	NEUTRAJ	0.0259	0.0201	0.0107	0.0283	0.0259	0.0251
	Traj2SimVec	0.0094	1.000	1.000	0.0249	0.0158	0.0110
$N=19$	NEUTRAJ	0.0262	0.0200	0.0103	0.0266	0.0259	0.0254
	Traj2SimVec	0.0094	1.000	1.000	0.0252	0.0158	0.0110

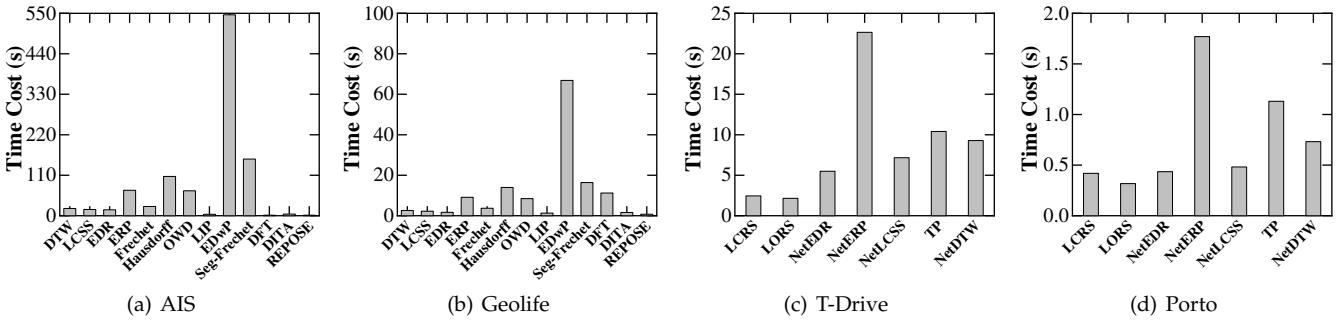


Fig. 15. Efficiency of Non-learning based Measures

TABLE 7
Robustness of Learning-based Measures vs. Noise (T-Drive)

Noise (%)	Measures	NetLCSS	NetDTW	TP	NetERP
$N=0$	GTS	0.2647	0.1439	0.2113	0.0869
	ST2Vec	0.1895	0.2691	0.3935	0.2570
$N=10$	GTS	0.0099	0.0097	0.0097	0.0098
	ST2Vec	0.0100	0.0101	0.0098	0.0102
$N=20$	GTS	0.0098	0.0101	0.0099	0.0101
	ST2Vec	0.0105	0.0103	0.0099	0.0103
$N=30$	GTS	0.0099	0.0102	0.0105	0.0102
	ST2Vec	0.0103	0.0103	0.0098	0.0098

TABLE 8
Robustness of Learning-based Measures vs. Noise (Porto)

Noise (%)	Measures	NetLCSS	NetDTW	TP	NetERP
$N=0$	GTS	0.2647	0.1439	0.2113	0.0869
	ST2Vec	0.2276	0.2928	0.3758	0.2505
$N=10$	GTS	0.0510	0.0501	0.0502	0.0509
	ST2Vec	0.0950	0.0354	0.0463	0.0340
$N=20$	GTS	0.0507	0.0506	0.0499	0.0501
	ST2Vec	0.0944	0.0334	0.0452	0.0348
$N=30$	GTS	0.0512	0.0495	0.0495	0.0494
	ST2Vec	0.0924	0.0348	0.0462	0.0313

respectively. As shown in Table 5 and Table 6, all similarity functions learned by NEUTRAJ and Traj2SimVec are robust to noises and the performance keeps stable when adding noises, indicating that NEUTRAJ and Traj2SimVec

TABLE 9
Efficiency of Learning-based Measures in Free Space (AIS)

Measures	NEUTRAJ		Traj2SimVec	
	$T(s)$	$Q(ms)$	$T(s)$	$Q(ms)$
DTW	125.01	46.80	135.19	6.59
LCSS	107.87	42.59	143.94	5.97
EDR	126.02	45.89	138.36	6.13
ERP	111.30	47.14	141.94	5.99
Frechet	104.01	42.80	136.37	7.46
Hausdorff	113.38	46.89	135.45	6.61

are noise-tolerant. This is an superior advantage over non-learning based measures. Based on Table 7 and Table 8, we obtain a similar observation as that of non-learning methods vs. noises, i.e., the performance of all similarity measures learned by ST2Vec and GTS significantly decrease after adding noises.

6.4 Efficiency Evaluation

Recall Table 2, we proceed to evaluate the efficiency of each measures in terms of Top- k trajectory similarity query task.

6.4.1 Non-learning based measures.

For each non-learning based measure, we calculate the average time cost of performing a Top-50 similarity search for one query trajectory. The results on four datasets when

TABLE 10
Efficiency of Learning-based Measures in Free Space (Geolife)

Measures	NEUTRAJ		Traj2SimVec	
	T(s)	Q(ms)	T(s)	Q(ms)
DTW	58.72	9.70	66.95	2.96
LCSS	59.25	9.64	65.83	3.36
EDR	54.16	9.01	66.58	3.35
ERP	57.64	9.43	65.85	3.94
Frechet	53.20	9.61	66.55	3.65
Hausdorff	57.80	9.60	63.30	3.59

TABLE 11
Efficiency of Learning-based Measures in Road Network (T-Drive)

Measures	GTS		ST2Vec	
	T(s)	Q(ms)	T(s)	Q(ms)
NetLCSS	8.62	2.67	171.84	3.61
NetDTW	8.66	2.66	166.04	2.56
TP	8.64	2.63	165.75	3.14
NetERP	9.83	2.82	212.68	12.32

using free space and network-oriented measures are shown in Figures 15, from which, we have the following insights.

As shown in Figure 15(a)–(b), the running times of DTW, LCSS, EDR, ERP, Frechet, OWD, and Hausdorff are similar. LIP shows high efficiency (i.e., the running time is around 4 seconds). EDwP shows low efficiency with more than 500 seconds running time. This is because the time complexity of non-learning based methods except LIP are $O(mn)$, LIP has a time complexity of $O((m+n)\log(m+n))$, and there are four dynamic processing tables during EdwP computation while others only have one table.

As depicted in Figure 15(c)–(d), the running times of LCRS, LORS, NetEDR, NetLCSS, and NetDTW are less than 10 seconds in T-Drive and 1 seconds in Porto, while those of TP and NetERP are much more. This is because TP additionally considers temporal information of trajectories, and NetERP requires computing the sum of distances between road intersections. Note that, for most standalone-based trajectory measures, we need to pre-compute the road network distance matrix, which takes more than two days. In contrast, LCRS and LORS compute the overlapping segments as inter-trajectory distance, which does not need the road network distance matrix. Overall, LCRS and LORS achieve the best efficiency in road network settings.

Finally, all of the distributed implementations (i.e., DFT, DITA, REPOSE) significantly improve the efficiency compared to the standalone implementation of Seg-Frechet and DTW. More specifically, among distributed implementations, REPOSE shows the best efficiency, because it leverages a novel heterogeneous global partitioning strategy to achieve load balancing. In addition, DFT shows the worst performance, because it traverses to find a pruning boundary before every query, while the others do not.

TABLE 12
Efficiency of Learning-based Measures in Road Network (Porto)

Measures	GTS		ST2Vec	
	T(s)	Q(ms)	T(s)	Q(ms)
NetLCSS	6.31	0.96	106.22	1.23
NetDTW	6.41	1.33	106.80	1.20
TP	6.35	1.08	119.15	1.15
NetERP	6.34	0.97	111.29	1.20

6.4.2 Learning-based measures.

For each learning-based measure, we report the model training and inference time of each epoch. Table 9, 10, 11, and 12 report the results (the training time T and the average query time cost for one query trajectory Q) of free space oriented models and road network oriented models on four datasets respectively. Based on these, we have the following insights.

As shown in Table 9 and Table 10, in free space settings, both the training time and query time under various similarity measures are similar. Compared with non-learning based measures, the average query time is significantly dropped by several orders of magnitude.

As shown in Table 11 and Table 12, in road network settings, ST2Vec takes more training time than GTS, while their query time is similar. This is because ST2Vec considers temporal information and computes the temporal similarity during model training. As observed, both the training time and query time of ST2Vec and GTS when learning most similarity measures are similar. However, when ST2Vec learns NetERP distance, it takes more training time and query time than other measures. This is because NetERP requires computing the sums of temporal distances between road intersections. Due to the fact that NetERP is designed only for spatial setting and it is a metric measure, it takes more time on temporal information when being expanded from spatial to spatio-temporal setting. Overall, NetERP is not suitable for spatio-temporal similarity analyses.

6.5 Scalability Evaluation

Finally, we explore the scalability of each measure and distributed implementation (if have) using length and cardinality parameters.

6.5.1 Non-learning based measures.

The results of standalone and distributed based methods are shown in Figures 16 and Figures 18 respectively, when varying trajectory length. Figures 17 and Figure 19 depict the results of standalone and distributed based methods, when varying cardinality of the dataset. Note that, DISON, which is the one and only distributed implementation for road network oriented measure, only supports threshold-based similarity query. For example, given a query trajectory Q , DISON requires a threshold (i.e., τ) and returns all trajectories whose similarity scores with Q satisfy τ . As DISON does not support Top- k similarity queries, we carefully tune the $\tau = 0.8$ so that it can return the same number of similar trajectories as other distributed methods. Finally, we report query time for standalone-based methods, and report both index and query time for distributed-based methods.

In terms of standalone-based methods, the first observation is that, for most standalone-based methods in Figure 16 and Figure 17, the query time increases significantly with the growth of trajectory length or data cardinality, which means that most measures are not suitable for large-scale processing. Also in Figure 17, with the growth of data cardinality, the query times of LIP, LCRS, and LORS increase slightly, among which, LORS shows stable performance when data cardinality varies from 60% to 100%.

In terms of distributed-based methods, they significantly reduce the time costs with the growth of data cardinality or

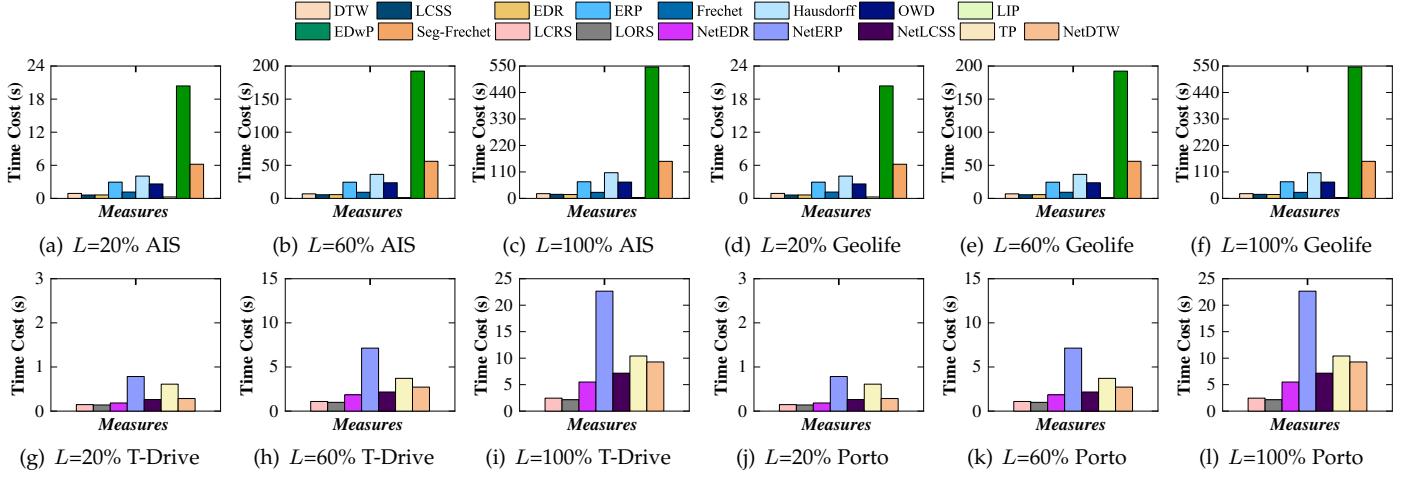


Fig. 16. Scalability Evaluation of Non-learning based Measures vs. Trajectory Length

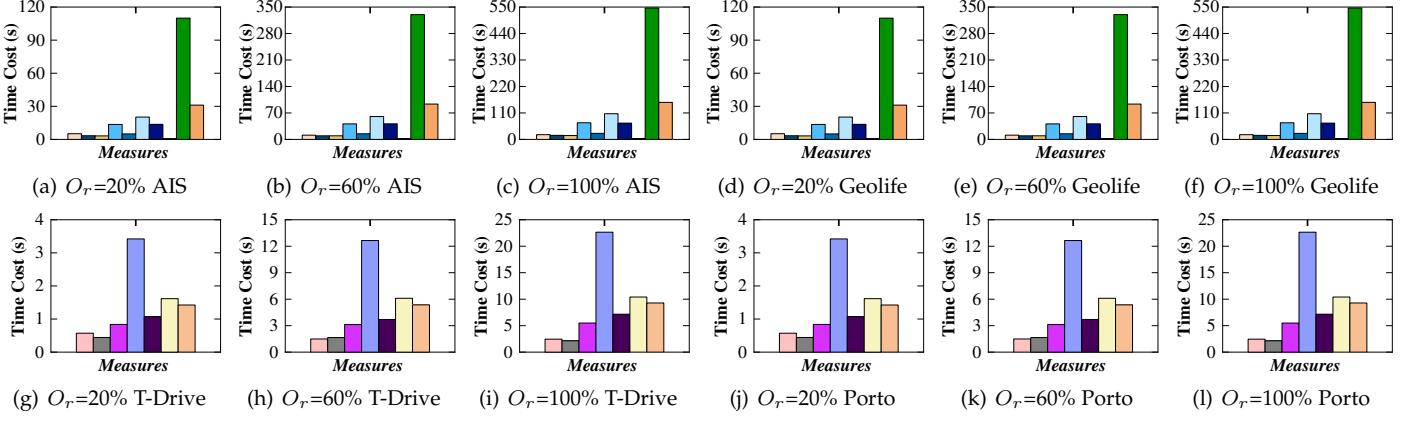


Fig. 17. Scalability Evaluation of Non-learning based Measures vs. Data Cardinality

TABLE 13
Scalability Evaluation of Learning-based Measures on Training Time and Query Time vs. Trajectory Length (AIS & T-Drive)

L(%)	L=20				L=60				L=100			
	NEUTRAJ		Traj2SimVec		NEUTRAJ		Traj2SimVec		NEUTRAJ		Traj2SimVec	
	T(s)	Q(ms)	T(s)	Q(ms)	T(s)	Q(ms)	T(s)	Q(ms)	T(s)	Q(ms)	T(s)	Q(ms)
Measures in Free Space												
DTW	101.12	42.19	45.36	4.85	114.06	43.72	91.92	6.82	125.01	46.80	135.19	6.59
LCSS	105.10	40.36	44.68	4.40	108.21	45.33	96.79	5.90	107.87	42.59	143.94	5.97
EDR	100.77	41.67	44.92	4.37	111.34	44.40	92.74	5.73	126.02	45.89	138.36	6.13
ERP	103.86	41.56	42.34	4.82	113.05	42.80	90.34	5.82	111.30	47.14	141.94	5.99
Frechet	103.86	41.33	44.89	4.80	114.11	41.48	96.56	6.83	104.01	42.80	136.37	7.46
Hausdorff	103.87	41.67	44.78	4.87	113.76	42.96	93.47	7.28	113.38	46.89	135.45	6.61
Measures in Road Network	GTS		ST2Vec		GTS		ST2Vec		GTS		ST2Vec	
	T(s)	Q(ms)	T(s)	Q(ms)	T(s)	Q(ms)	T(s)	Q(ms)	T(s)	Q(ms)	T(s)	Q(ms)
NetLCSS	3.08	0.79	27.13	0.81	5.84	2.21	84.69	1.93	8.62	2.67	171.84	3.61
NetDTW	2.98	0.77	25.58	0.82	5.35	1.36	87.90	1.97	8.66	2.66	166.04	2.56
TP	3.06	0.65	27.21	0.92	5.29	1.71	84.04	1.97	8.64	2.63	165.75	3.14
NetERP	3.06	0.81	26.17	0.94	5.45	1.69	82.56	1.66	8.83	2.82	212.68	12.32

trajectory length. In free space, as shown in Figures 18(a)–(b), (d)–(e) and 19(a)–(b), (d)–(e), the query times and indexing times of DITA and REPOSE vary gently. It is obvious that distributed techniques can improve the scalability of trajectory measures, where REPOSE performs the best. Relatively, DFT takes more time when data amount and trajectory length grow, which is slightly inferior. This is because, DFT uses R-tree for indexing while others use trie-like index and R-tree. Trie-like index enables to compute accumulative distance level by level, which performs better than R-tree based index. In road networks, as shown in

Figures 18(c), (f) and 19(c), (f), the query time of DISON is less than 15 seconds, and the time cost slightly increases with the growth of cardinality and trajectory length.

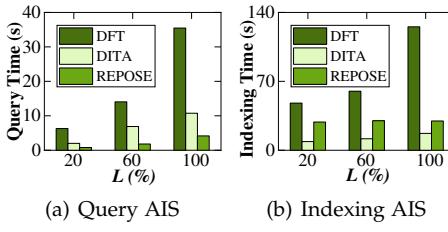
6.5.2 Learning-based methods.

For simplicity, we mainly evaluate the scalability performance of learning-based methods on AIS and T-Drive. Tables 13 and 14 show learning-based measures when varying cardinality and length. Overall, we have four observations. First, the time costs of all measures slightly increase with the growth of cardinality or length, while the training times

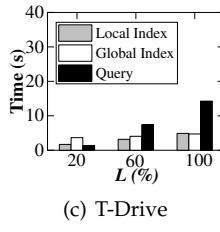
TABLE 14

Scalability Evaluation of Learning-based Measures on Training Time and Query Time vs. Data Cardinality (AIS & T-Drive)

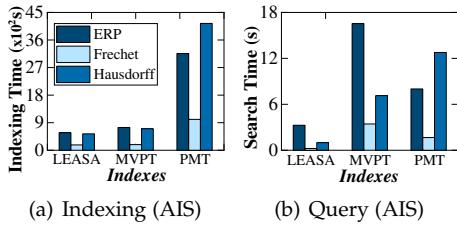
$O_r(\%)$	$O_r=20$				$O_r=60$				$O_r=100$			
	NEUTRAJ		Traj2SimVec		NEUTRAJ		Traj2SimVec		NEUTRAJ		Traj2SimVec	
Measures in Free Space	$T(s)$	$Q(ms)$	$T(s)$	$Q(ms)$	$T(s)$	$Q(ms)$	$T(s)$	$Q(ms)$	$T(s)$	$Q(ms)$	$T(s)$	$Q(ms)$
DTW	100.92	18.31	45.26	1.30	122.88	29.11	106.01	3.32	125.01	46.80	135.19	6.59
LCSS	109.90	18.26	29.64	1.28	125.88	29.26	107.02	3.39	107.87	42.59	143.94	5.97
EDR	100.79	18.08	44.52	1.33	123.67	29.69	106.56	3.21	126.02	25.89	138.36	6.13
ERP	101.95	18.48	45.35	1.33	123.46	29.56	106.58	3.46	111.30	47.14	141.94	5.99
Frechet	102.10	18.32	43.69	1.32	126.28	29.26	107.24	4.55	104.01	42.80	136.37	7.46
Hausdorff	101.10	18.39	44.58	1.30	122.48	29.20	106.65	3.34	113.38	46.89	135.45	6.61
Measures in Road Network	GTS		ST2Vec		GTS		ST2Vec		GTS		ST2Vec	
	$T(s)$	$Q(ms)$	$T(s)$	$Q(ms)$	$T(s)$	$Q(ms)$	$T(s)$	$Q(ms)$	$T(s)$	$Q(ms)$	$T(s)$	$Q(ms)$
NetLCSS	1.46	0.41	39.70	0.64	4.71	1.46	84.95	1.83	8.62	2.67	171.84	3.61
NetDTW	1.49	0.50	28.98	0.52	4.72	1.28	84.58	1.52	8.66	2.66	166.04	2.56
TP	1.49	0.51	27.07	0.50	4.66	1.28	86.04	2.04	8.64	2.63	165.75	3.14
NetERP	1.49	0.52	30.40	0.59	5.33	1.24	90.59	1.88	8.83	2.82	212.68	12.32



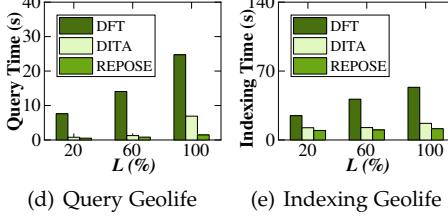
(a) Query AIS (b) Indexing AIS



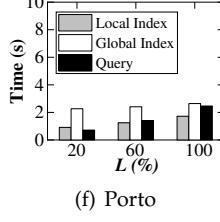
(c) T-Drive



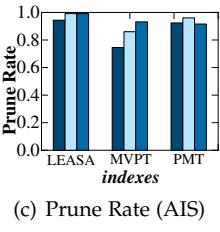
(a) Indexing (AIS) (b) Query (AIS) (c) Prune Rate (AIS)



(d) Query Geolife (e) Indexing Geolife

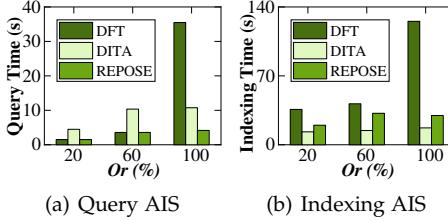


(f) Porto

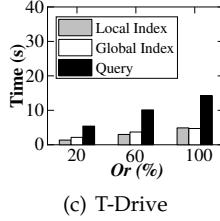


(d) Indexing (Geolife) (e) Query (Geolife) (f) Prune Rate (Geolife)

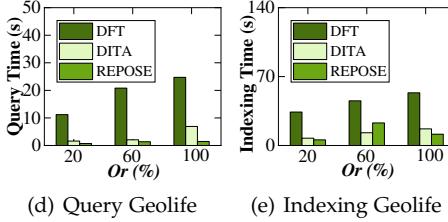
Fig. 18. Scalability Evaluation of Distributed-based Measures vs. Trajectory Length



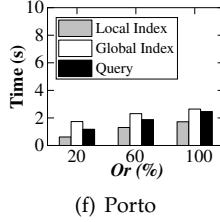
(a) Query AIS (b) Indexing AIS



(c) T-Drive



(d) Query Geolife (e) Indexing Geolife



(f) Porto

Fig. 19. Scalability Evaluation of Distributed-based Measures vs. Data Cardinality

grow significantly. Because it takes more time to capture the information of more sampling points for model training, while query (inference) only applies the well-trained model to obtain the Top- k results in a constant time. Second, when varying cardinality or trajectory length, DTW, ERP, Frechet, Hausdorff, and LCSS have stable time costs. Third, the time costs of learning-based measures are much lower than that of non-learning based measures, even compared with the distributed-based measures. Finally, the time to

learn different measures is about the same, indicating that learning-based measures are efficient to accelerate a broad range of measures.

6.6 Case Study

Recall Section 3, the metric measures can be used for indexing by applying the triangle inequality, which helps to achieve more efficient similarity search. In this subsection, we evaluate the efficiency performance of three metric measures (i.e., ERP, Frechet and Hausdorff) mentioned aforementioned, when trajectories are indexed.

We select three representative pivot-based indexing structures (i.e., LEASA [42], MVP-Tree (MVPT) [6] and PMT-Tree (PMT) [58]) which can be used for triangle inequality pruning [11]. For each metric measure and indexing structure, we first select several pivot trajectories (i.e., P_1, \dots, P_i, \dots) from all trajectories using HF method [10], computing the measure distances among each pivot trajectory and all of the rest trajectories. Then, given a query trajectory Q and a threshold d , we calculate the measure distances between Q and each pivot trajectory. Finally, the triangle inequality pruning strategy is employed, pruning off all trajectories T_i in which $|measure(Q, P_i) - measure(P_i, T_i)| > d$. In this subsection, We apply LEASA, MVPT and PMT to index trajectories, and use ERP, Frechet and Hausdorff to perform triangle inequality pruning. In addition, we carefully tune the

pivot number as 5 to balance the indexing time and pruning effect (i.e., computed by $\text{PruneRate} = \frac{\text{TrajNum}_{\text{pruned}}}{\text{TrajNum}_{\text{all}}}$).

Figure 20 shows the efficiency and pruning effect of various indexing structures on the three measures. And we have the following insights: (i) The indexing and pruning techniques improve the efficiency of all metric measures according to Figure 15(a)–(b). The search time cost is related to PruneRate , where the higher PruneRate is, the more efficiency performance improved. (ii) The indexing and pruning techniques are able to change the efficiency ranking among measures. As shown in Figure 15, in terms of metric measures, Frechet has the best efficiency performance, and Hausdorff performs worse. However, the efficiency ranking changes when applying different indexing structures. For example, Hausdorff performs better efficiency than ERP when using LEASA and MVPT in AIS (see Figure 20)(b). For another example, ERP performs the best when using LEASA and PMT in Geolife (see Figure 20)(e). (iii) The indexing time is much higher than search. Moreover, the indexing and pruning time of different measures on the same indexing structure are different. As shown in Figure 20, Frechet spends the less time on indexing and pruning, while keeps stable and high PruneRate . Therefore, compared with non-metric measures, metric indexing structures help to make up for the efficiency defects of metric measures. And the relative efficiency of metric measures can be changed by introducing metric indexing structures and triangle inequality pruning strategy. Thus, it is meaningful to consider whether a measure is metric or not when selecting a similarity measure.

6.7 Summary

In terms of non-learning measures, free space measures Frechet and Hausdorf, as well as road-network measures LORS and LCRS perform better than other measures in their respective categories. However, some classic measures (e.g., ERP) have poor performance, especially when datasets are with large spatial span and uneven distribution. In addition, road-network measures, e.g., NetEDR and NetERP, that directly extend free space measures into road networks by replacing the Euclidean distance with the road network distance perform poor, and thus, users should design road-network aware measures when they deal with vehicle/people trajectory analyses. Moreover, almost all the existing road-network-oriented measures are sensitive to noises, which is a potential study direction.

In terms of learning-based measures, compared with classic handcrafted measures, although they can achieve much higher robustness, efficiency, and scalability, the effectiveness is insufficient, which limits their applications and is another future direction for effective/interpretable AI-driven similarity analysis.

Last but not the least, distributed techniques have shown great improvement for non-learning and free-space measures, but there is still a huge research gap to be filled between distributed processing and AI-driven/Network-constrained trajectory similarity analysis, which also deserves to be explored in the future.

7 FUTURE WORK

Even with a large amount of research work on spatio-temporal trajectory similarity/distance measure, there are a huge set of challenges to be addressed and gaps to be filled. In this section, we suggest some potential future research directions based on the experimental results.

(i) **Effective AI-driven similarity analysis.** As observed in Section 6, learning based measures have greatly accelerated the similarity computation of non-learning measures. However, existing learning based measures has poor effectiveness to deal with trajectories with large spatial span and uneven distribution. And the aprroximation process of deel learning models is unexplainable because the nonlinear features in neural networks. Thus, how to improve the capability and interpretable of deep learning models to approximate distance functions without loss of efficiency is an urgent challenge.

(ii) **Distributed trajectory similarity analysis.** Distributed techniques have shown promising capability to boost non-learning based similarity measures in free space. However, there is a lack of work in distributed similarity computation in road network settings. When designing trajectory similarity measure in road network, researchers tend to find the shortest path in a graph as the road network distance. This process is very time consuming with $O(N^2)$ time complexity, where N denotes the number of road vertices. Thus, distributed trajectory similarity measure in road network, which is discussed in Section 3.2, is still an open issue to be explored. There is only DISON [82] searching for distributed similarity measure in road network. DISON uses a new measure named LCRS to find overlapping road segments, which need not to compute the road network distance in the graph. However, LCRS is not suitable for all scenarios. For example, when considering temporal information of trajectory data, there is no timestamp defined on road segments. Therefore, an efficiency method to deal with road network distance computation is still urgently needed. And accelerating similarity computing process in road network with the help of distributed techniques is a potential research topic.

(iii) **Distributed learning-based trajectory measures.** As observed in Section 6, though a deep learning model spends little time on similarity search query, it takes quite a bit of time to train the model. And the training time increases obviously as the data cardinality and trajectory length grow. However, distributed techniques can help to train a model in parallel and reduce the training time, where the main challenge lies on how to partition the data or model to achieve load balancing. In addition, though some researchers has explored distributed deep learning methods for several applications [24], [39], [67], there is little achievement in trajectory analytics. Thus, It is a possible research issue for various applications (e.g., online trajectory analytics) in the future.

8 CONCLUSIONS

In this paper, we review trajectory distance measures from three-dimensional perspectives. Then, we offer an evaluation benchmark to examine the effectiveness, robustness, efficiency, and scalability of each measure. According to

experimental results, we give objective insights for trajectory measure selection in varying scenarios. Based on the issues to be addressed in the community, we believe effective/interpretable AI-driven similarity analysis as well as distributed trajectory similarity analyses in road networks or high-dimensional embedding space, are promising directions.

REFERENCES

- [1] O. Abul, F. Bonchi, and M. Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *ICDE*, pages 376–385, 2008.
- [2] H. Alt and M. Godau. Computing the fréchet distance between two polygonal curves. *Int. J. Comput. Geom. Appl.*, 5:75–91, 1995.
- [3] G. L. Andrienko, N. V. Andrienko, and S. Wrobel. Visual analytics tools for analysis of movement data. *SIGKDD*, 9(2):38–46, 2007.
- [4] S. Atev, G. Miller, and N. P. Papanikopoulos. Clustering of vehicle trajectories. *TITS*, 11(3):647–657, 2010.
- [5] N. M. Blachman. On fourier series for gaussian noise. *Inf. Control.*, 1(1):56–63, 1957.
- [6] T. Bozkaya and Z. M. Özsoyoglu. Distance-based indexing for high-dimensional metric spaces. In *SIGMOD*, pages 357–368, 1997.
- [7] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk. On map-matching vehicle tracking data. In *PVLDB*, pages 853–864, 2005.
- [8] W. Cao, Z. Wu, D. Wang, J. Li, and H. Wu. Automatic user identification method across heterogeneous mobility data sources. In *ICDE*, pages 978–989, 2016.
- [9] L. Chen, Y. Gao, Z. Fang, X. Miao, C. S. Jensen, and C. Guo. Real-time distributed co-movement pattern detection on streaming trajectories. *VLDB*, 12(10):1208–1220, 2019.
- [10] L. Chen, Y. Gao, X. Li, C. S. Jensen, and G. Chen. Efficient metric indexing for similarity search. In *ICDE*, pages 591–602, 2015.
- [11] L. Chen, Y. Gao, X. Song, Z. Li, X. Miao, and C. S. Jensen. Indexing metric spaces for exact similarity search. *CoRR*, abs/2005.03468, 2020.
- [12] L. Chen and R. T. Ng. On the marriage of l_p -norms and edit distance. In *VLDB*, pages 792–803, 2004.
- [13] L. Chen, M. T. Özsu, and V. Oria. Robust and fast similarity search for moving object trajectories. In *SIGMOD*, pages 491–502, 2005.
- [14] Q. Chen, L. Chen, X. Lian, Y. Liu, and J. X. Yu. Indexable PLA for efficient similarity search. In *VLDB*, pages 435–446, 2007.
- [15] R. S. de Sousa, A. Boukerche, and A. A. F. Loureiro. Vehicle trajectory similarity: Models, methods, and applications. *ACM Comput. Surv.*, 53(5):94:1–94:32, 2020.
- [16] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. J. Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *PVLDB*, 1(2):1542–1552, 2008.
- [17] S. Dodge, P. Laube, and R. Weibel. Movement similarity assessment using symbolic representation of trajectories. *GIS*.
- [18] Z. Fang, Y. Du, L. Chen, Y. Hu, Y. Gao, and G. Chen. E^2 dtc: An end to end deep trajectory clustering framework via self-training. In *ICDE*, pages 696–707, 2021.
- [19] Z. Fang, Y. Du, X. Zhu, D. Hu, L. Chen, Y. Gao, and C. S. Jensen. Spatio-temporal trajectory similarity learning in road networks. In *SIGKDD*, pages 347–356, 2022.
- [20] Z. Fang, L. Pan, L. Chen, Y. Du, and Y. Gao. MDTP: A multi-source deep traffic prediction framework over spatio-temporal trajectory data. *PVLDB*, 14(8):1289–1297, 2021.
- [21] E. Frentzos, K. Gratsias, and Y. Theodoridis. Index-based most similar trajectory search. In *ICDE*, pages 816–825, 2007.
- [22] B. Friedrich, B. Cauchi, A. Hein, and S. J. F. Fudickar. Transportation mode classification from smartphone sensors via a long-short-term-memory network. *CoRR*, abs/1910.04739, 2019.
- [23] A. Gasmelseed and N. H. Mahmood. Study of hand preferences on signature for right-handed and left-handed peoples. *Int. J. Adv. Eng. Technol.*, 2011.
- [24] R. Gu, Y. Chen, S. Liu, H. Dai, G. Chen, K. Zhang, Y. Che, and Y. Huang. Liquid: Intelligent resource estimation and network-efficient scheduling for deep learning jobs on distributed GPU clusters. *IEEE TPDS*, 33(11):2808–2820, 2022.
- [25] R. H. Güting, T. Behr, and J. Xu. Efficient k -nearest neighbor search on moving object trajectories. *VLDBJ*, 19(5):687–714, 2010.
- [26] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *SIGMOD*, pages 47–57, 1984.
- [27] P. Han, J. Wang, D. Yao, S. Shang, and X. Zhang. A graph-based approach for trajectory similarity computation in spatial networks. In *SIGKDD*, pages 556–564, 2021.
- [28] X. Huang, Y. Yin, S. Lim, G. Wang, B. Hu, J. Varadarajan, S. Zheng, A. Bulusu, and R. Zimmermann. Grab-posisi: An extensive real-life GPS trajectory dataset in southeast asia. In *SIGSPATIAL*, pages 1–10, 2019.
- [29] C. Hung, W. Peng, and W. Lee. Clustering and aggregating clues of trajectories for mining trajectory patterns and routes. *VLDBJ*, 24(2):169–192, 2015.
- [30] E. J. Keogh and M. J. Pazzani. Scaling up dynamic time warping for datamining applications. In *SIGKDD*, pages 285–289, 2000.
- [31] S. Koide, C. Xiao, and Y. Ishikawa. Fast subtrajectory similarity search in road networks under weighted edit distance constraints. *PVLDB*, 13(11):2188–2201, 2020.
- [32] I. Kontopoulos, A. Makris, D. Zissis, and K. Tserpes. A computer vision approach for trajectory classification. In *MDM*, pages 163–168, 2021.
- [33] J. Lee, J. Han, and K. Whang. Trajectory clustering: a partition-and-group framework. In *SIGMOD*, pages 593–604, 2007.
- [34] S. T. Leutenegger, J. M. Edgington, and M. A. López. STR: A simple and efficient algorithm for r-tree packing. In *ICDE*, pages 497–506, 1997.
- [35] T. Li, L. Chen, C. S. Jensen, and T. B. Pedersen. TRACE: real-time compression of streaming trajectories in road networks. *PVLDB*, 14(7):1175–1187, 2021.
- [36] X. Li, J. Han, S. Kim, and H. Gonzalez. ROAM: rule- and motif-based anomaly detection in massive moving object data sets. In *SIAM*, pages 273–284, 2007.
- [37] X. Li, K. Zhao, G. Cong, C. S. Jensen, and W. Wei. Deep representation learning for trajectory similarity computation. In *ICDE*, pages 617–628, 2018.
- [38] Z. Li, J. Han, M. Ji, L. Tang, Y. Yu, B. Ding, J. Lee, and R. Kays. Movemine: Mining moving object data for discovery of animal movement patterns. *TIST*, 2(4):37:1–37:32, 2011.
- [39] T. Liu, T. Miao, Q. Wu, Z. Li, G. He, J. Wu, S. Zhang, X. Yang, G. Tyson, and G. Xie. Modeling and optimizing the scaling performance in distributed deep learning training. In *WWW*, pages 1764–1773, 2022.
- [40] S. Ma, Y. Zheng, and O. Wolfson. T-share: A large-scale dynamic taxi ridesharing service. In *ICDE*, pages 410–421, 2013.
- [41] J. D. Mazimpaka and S. Timpf. Trajectory data mining: A review of methods and applications. *J. Spatial Inf. Sci.*, 13(1):61–99, 2016.
- [42] L. Micó, J. Oncina, and E. Vidal. A new version of the nearest-neighbour approximating and eliminating search algorithm (AES) with linear preprocessing time and memory requirements. *Pattern Recognit. Lett.*, 15(1):9–17, 1994.
- [43] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013.
- [44] M. F. Mokbel, C. Chow, and W. G. Aref. The new casper: Query processing for location services without compromising privacy. In *VLDB*, pages 763–774, 2006.
- [45] C. Myers, L. R. Rabiner, and A. E. Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Trans. Acous. Speech Signal Process.*, 28(6):623–635, 1980.
- [46] K. S. Naveh and J. Kim. Urban trajectory analytics: Day-of-week movement pattern mining using tensor factorization. *TITS*, 20(7):2540–2549, 2019.
- [47] OpenStreetMap. <https://master apis.dev.openstreetmap.org/>. 2022.
- [48] N. Pelekis, I. Kopanakis, G. Marketos, I. Ntoutsi, G. L. Andrienko, and Y. Theodoridis. Similarity search in trajectory databases. In *TIME*, pages 129–140, 2007.
- [49] A. project. <https://marinecadastre.gov/ais/>. 2019.
- [50] T.-D. Project. <https://www.microsoft.com/en-us/research/publication/t-drive-trajectorydatasample/>. 2008.
- [51] S. Ranu, D. P. A. D. Telang, P. Deshpande, and S. Raghavan. Indexing and matching trajectories under inconsistent sampling rates. In *ICDE*, pages 999–1010, 2015.
- [52] Y. S. Resheff. Online trajectory segmentation and summary with applications to visualization and retrieval. In *IEEE BigData*, pages 1832–1840, 2016.

- [53] M. T. Robinson. The temporal development of collision cascades in the binary collision approximation. *Nucl. Instrum. Methods Phys. Res. Sect. B*, 48(1-4):408–413, 1990.
- [54] S. Salvador and P. Chan. Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.*, 11(5):561–580, 2007.
- [55] A. C. Sanderson and A. K. C. Wong. Pattern trajectory analysis of nonstationary multivariate data. *IEEE Trans. Syst. Man Cybern.*, 10(7):384–392, 1980.
- [56] S. Shang, L. Chen, Z. Wei, C. S. Jensen, K. Zheng, and P. Kalnis. Trajectory similarity join in spatial networks. *PVLDB*, 10(11):1178–1189, 2017.
- [57] Z. Shang, G. Li, and Z. Bao. DITA: distributed in-memory trajectory analytics. In *SIGMOD*, pages 725–740, 2018.
- [58] T. Skopal, J. Pokorný, and V. Snásel. Pm-tree: Pivoting metric tree for similarity search in multimedia databases. In *ADBIS*, 2004.
- [59] H. Su, S. Liu, B. Zheng, X. Zhou, and K. Zheng. A survey of trajectory distance measures and performance evaluation. *VLDBJ*, 29(1):3–32, 2020.
- [60] Y. Suzuki, J. Ishizuka, and K. Kawagoe. A similarity search of trajectory data using textual information retrieval techniques. In *DASFAA*, volume 4947, pages 627–634, 2008.
- [61] N. Ta, G. Li, Y. Xie, C. Li, S. Hao, and J. Feng. Signature-based trajectory similarity join. *TKDE*, 29(4):870–883, 2017.
- [62] D. A. Tedjopurnomo, X. Li, Z. Bao, G. Cong, F. M. Choudhury, and A. K. Qin. Similar trajectory search with spatio-temporal deep representation learning. *TIST*, 12(6):77:1–77:26, 2021.
- [63] Y. Tong, Z. Zhou, Y. Zeng, L. Chen, and C. Shahabi. Spatial crowdsourcing: a survey. *VLDBJ*, 29(1):217–250, 2020.
- [64] K. Toohey and M. Duckham. Trajectory similarity measures. *ACM SIGSPATIAL*, 7(1):43–50, 2015.
- [65] Uber. <https://expandedramblings.com/index.php/uber-statistics/>. 2022.
- [66] M. Vlachos, D. Gunopoulos, and G. Kollios. Discovering similar multidimensional trajectories. In *ICDE*, pages 673–684, 2002.
- [67] L. Wang, Q. Luo, and S. Yan. DIESEL+: accelerating distributed deep learning tasks on image datasets. *IEEE TPDS*, 33(5):1173–1184, 2022.
- [68] S. Wang, Z. Bao, J. S. Culpepper, and G. Cong. A survey on trajectory data management, analytics, and learning. *ACM Comput. Surv.*, 54(2):39:1–39:36, 2021.
- [69] S. Wang, Z. Bao, J. S. Culpepper, Z. Xie, Q. Liu, and X. Qin. Torch: A search engine for trajectory data. In *SIGIR*, pages 535–544, 2018.
- [70] S. Wang, J. Cao, and P. S. Yu. Deep learning for spatio-temporal data mining: A survey. *CoRR*, abs/1906.04928, 2019.
- [71] S. Wang, Y. Sun, and Z. Bao. On the efficiency of k-means clustering: Evaluation, optimization, and algorithm selection. *PVLDB*, 14(2):163–175, 2020.
- [72] T. U. Wien, T. Eiter, T. Eiter, H. Mannila, and H. Mannila. Computing discrete fréchet distance. *Technical report, Citeseer*, 64(3):636–637, 1994.
- [73] D. Xie, F. Li, and J. M. Phillips. Distributed trajectory similarity search. *PVLDB*, 10(11):1478–1489, 2017.
- [74] X. Xie, W. Philips, P. Veelaert, and H. K. Aghajan. Road network inference from GPS traces using DTW algorithm. In *ITSC*, pages 906–911, 2014.
- [75] C. Yang and G. Gidófalvi. Fast map matching, an algorithm integrating hidden markov model with precomputation. *Int. J. Geogr. Inf. Sci.*, 32(3):547–570, 2018.
- [76] P. Yang, H. Wang, D. Lian, Y. Zhang, L. Qin, and W. Zhang. TMN: trajectory matching networks for predicting similarity. In *ICDE*, pages 1700–1713, 2022.
- [77] P. Yang, H. Wang, Y. Zhang, L. Qin, W. Zhang, and X. Lin. T3S: effective representation learning for trajectory similarity computation. In *ICDE*, pages 2183–2188, 2021.
- [78] D. Yao, G. Cong, C. Zhang, and J. Bi. Computing trajectory similarity in linear time: A generic seed-guided neural metric learning approach. In *ICDE*, pages 1358–1369, 2019.
- [79] D. Yao, H. Hu, L. Du, G. Cong, S. Han, and J. Bi. Trajgat: A graph-based long-term dependency modeling approach for trajectory similarity computation. In *SIGKDD*, pages 2275–2285, 2022.
- [80] D. Yao, C. Zhang, Z. Zhu, J. Huang, and J. Bi. Trajectory clustering via deep representation learning. In *IJCNN*, pages 3880–3887, 2017.
- [81] B. Yi, H. V. Jagadish, and C. Faloutsos. Efficient retrieval of similar time sequences under time warping. In *ICDE*, pages 201–208, 1998.
- [82] H. Yuan and G. Li. Distributed in-memory trajectory similarity search and join on road network. In *ICDE*, pages 1262–1273, 2019.
- [83] H. Yuan, G. Li, Z. Bao, and L. Feng. Effective travel time estimation: When historical trajectories over road networks matter. In *SIGMOD*, 2020.
- [84] J. Yuan, Y. Zheng, C. Zhang, X. Xie, and G. Sun. An interactive-voting based map matching algorithm. In *MDM*, pages 43–52, 2010.
- [85] D. Zhang, Z. Chang, S. Wu, Y. Yuan, and G. Chen. Continuous trajectory similarity search for online outlier detection. *IEEE TKDE*, PP(99):1–1, 2020.
- [86] H. Zhang, X. Zhang, Q. Jiang, B. Zheng, Z. Sun, W. Sun, and C. Wang. Trajectory similarity learning with auxiliary supervision and optimal matching. In *IJCAI*, pages 3209–3215, 2020.
- [87] B. Zheng, L. Weng, X. Zhao, K. Zeng, X. Zhou, and C. S. Jensen. REPOSE: distributed top-k trajectory similarity search with local reference point tries. In *ICDE*, pages 708–719, 2021.
- [88] K. Zheng, Y. Zheng, X. Xie, and X. Zhou. Reducing uncertainty of low-sampling-rate trajectories. In *ICDE*, pages 1144–1155, 2012.
- [89] Y. Zheng. Trajectory data mining: An overview. *TIST*, 6(3):29:1–29:41, 2015.
- [90] Y. Zheng, L. Liu, L. Wang, and X. Xie. Learning transportation mode from raw gps data for geographic applications on the web. In *WWW*, pages 247–256, 2008.
- [91] Y. Zheng, L. Wang, R. Zhang, X. Xie, and W. Ma. Geolife: Managing and understanding your past life over maps. In *MDM*, pages 211–212, 2008.
- [92] X. Zuo and X. Jin. General hierarchical model (GHM) to measure similarity of time series. *SIGMOD*, 36(1):13–18, 2007.
- [93] X. Zuo and X. Jin. General hierarchical model (GHM) to measure similarity of time series. *SIGMOD Rec.*, 36(1):13–18, 2007.

Danlei Hu received the BS degree in computer science from China Agricultural University, China, in 2021. She is currently working toward the PhD degree in the College of Computer Science, Zhejiang University, China. Her research interests include trajectory data mining and analytics.



Lu Chen received the PhD degree in computer science from Zhejiang University, China, in 2016. She was an assistant professor in Aalborg University for a 2-year period from 2017 to 2019, and she was an associated professor in Aalborg University for a 1-year period from 2019 to 2020. She is currently a ZJU Plan 100 Professor in the College of Computer Science, Zhejiang University, Hangzhou, China. Her research interests include indexing and querying metric spaces, graph databases, and database usability.



Hanxi Fang is currently a senior undergraduate student in Zhejiang University, Hangzhou, China. She is pursuing the B.S. degree in Geography Information Science. Her current research interests include artificial intelligence and spatio-temporal data mining.





Ziquan Fang received the BS degree in computer science from China Agricultural University, China, in 2018. He is currently working toward the PhD degree in the College of Computer Science, Zhejiang University, China. His research interests include spatio-temporal databases, distributed and AI-enhanced trajectory data management and analytics.



Yunjun Gao received the PhD degree in computer science from Zhejiang University, China, in 2008. He is currently a professor in the College of Computer Science, Zhejiang University, China. His research interests include spatial and spatio-temporal databases, metric and incomplete/uncertain data management, graph databases, spatio-textual data processing, and database usability. He is a member of the ACM and the IEEE.