UNIVERSITY OF CALIFORNIA

Santa Barbara

Explicit and implicit movement similarity and their applications in movement analysis

A Thesis submitted in partial satisfaction of the
requirements for the degree Master of Arts
in Geography

by

Zijian Wan

Committee in charge:

Professor Somayeh Dodge, Chair

Professor Konstadinos Goulias

Professor Ambuj Singh

June 2022

The thesis of Zijian Wan is approved.

Konstadinos Goulias

Ambuj Singh

Somayeh Dodge, Committee Chair

June 2022

Explicit and implicit movement similarity and their applications in movement analysis

by

Zijian Wan

To my parents,

Guoqiao Wan and Bo Peng.

Acknowledgements

My gratitude goes first to my advisor, Prof. Somayeh Dodge, for her invaluable advice and continuous support throughout my first years as a graduate student at UC Santa Barbara. My graduate program started in an unprecedently challenging time amid the COVID pandemic. Her support and thoughtfulness are the light that guides me through the darkness. It is my honor to work with Prof. Dodge.

Also, my appreciation extends to my committee members, Prof. Konstadinos Goulias and Prof. Ambuj Singh, for their expert suggestions in helping me revise this thesis.

Finally, I would like to thank my family, friends, and colleagues, especially all members of the MOVE Lab, for all their encouragement and support throughout my studies.

Abstract

Explicit and implicit movement similarity and their applications in movement analysis

by

Zijian Wan

Movement similarity is a hot topic and the foundation of a plethora of methodologies in movement analysis. In this thesis, movement similarity is classified into two categories. The first category, explicit movement similarity, is defined as the closeness of trajectories in space and time. It is directly quantified along the path of individuals using a wide variety of trajectory similarity measures, such as Fréchet distance and dynamic time warping. On the other hand, the second category, *implicit movement similarity,* is defined as the consistency in movement and behavioral patterns of an individual or a group of moving entities. It can serve as a high-level representation of movement patterns of an individual or a group of moving entities and can be applied to solve many types of movement-related problems, such as trajectory prediction and interpolation. Surrounding the topic of movement similarity, this thesis investigates two methodologies based on the explicit and implicit movement similarity, respectively, to demonstrate their applications in movement analysis.

Explicit movement similarity in this thesis is utilized to unravel the associations between migration paths and underlying environmental correlates influencing movement choices of migratory turkey vultures *(Cathartes aura)* in North America. Multiple commonly used trajectory-similarity measures including Fréchet distance, dynamic time warping (DTW), Hausdorff distance, longest common subsequence (LCSS), and edit distance are integrated into a hierarchical clustering approach to identify variations in turkey vultures' migration path choices over multiple seasons. At each hierarchy, the optimal clustering setting, i.e., a distance metric together with the number of clusters, is selected automatically based on the silhouette coefficient. Using 15 years of tracking data of turkey vultures during their fall and spring migration seasons, seasonal clusters are identified and then annotated with environmental variables for Kolmogorov-Smirnov (KS) test and Jensen-Shannon distance (JSD) calculation to examine the variation between clusters and the background and variation between clusters.

In terms of the application of implicit movement similarity, this thesis proposes a trajectory interpolation model with an encoder-decoder architecture based on gated recurrent units (GRUs) to interpolate trajectory gaps (missing values). The proposed model is able to read a trajectory containing a gap in both chronological and reverse chronological orders. The information obtained from these two directions is fused to learn the implicit movement similarity contained in that trajectory, which is later used to reconstruct the complete trajectory with the original gap filled in. The proposed interpolation method is validated using turkey vulture migration trajectories. Interpolation results demonstrate that the proposed method is capable of capturing implicit movement similarity from trajectories for interpolation purposes, since without which, some gaps would be difficult to interpolate accurately using traditional interpolation methods.

Movement similarity is a promising field in computational movement analysis and is often used as a foundation of other machine learning and modeling methodologies such as trajectory classification, behavioral model detection, and movement prediction. On the foundation of explicit and implicit movement similarity, researchers can build multifarious models for diverse moving entities, such as animals, humans, vessels, and vehicles.

Contents

List of Figures

Chapter 1

Introduction

Movement is a continuous phenomenon, while in general, it is recorded as discrete snapshots at a certain temporal resolution. Those snapshots, usually called tracking points, have three components: spatial information, e.g., longitude and latitude, timestamp, and other attributes of the moving entity or its behavioral and environmental context information. When the temporally adjacent tracking points of the same moving entities are connected, these snapshots are weaved together to become trajectories (see Figure 1 for an example), which are the fundamental building blocks of computational movement analysis (Laube, 2014; Tao et al., 2021).

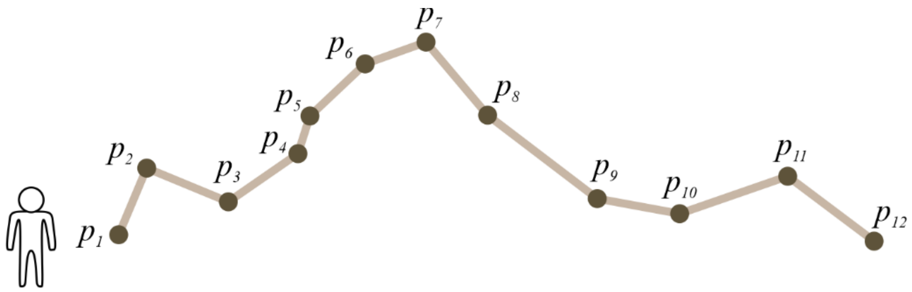


Figure 1. A schematic illustration of a trajectory

The recent two decades have seen advances in tracking technologies as well as in techniques related to big data processing and storage. These advances result in the availability of geographic information recording the movement of moving entities (e.g., animals, humans, vehicles, vessels) with not only high resolution but also massive data volume (Laube, 2014). These relatively novel and naturally spatiotemporal movement data inherently hold big-data properties, including velocity, volume, value, variety, and veracity (Demchenko et al., 2014). With such sufficient data recording the movement of the same or similar moving entities over a period of time, some inspiring questions that emerge intuitively are: How similar are the movements of two or more similar or related entities? How similar does an individual move in different time periods?... Pointing at movement similarity, these are important questions in geographic information science (GIScience) and in movement analysis in particular (Laube et al., 2007; Ranacher & Tzavella, 2014; Vlachos et al., 2004). Movement similarity is a promising field and research direction in computational movement analysis, and it is often used as a foundation of other machine learning and modeling methodologies, such as inferring movement patterns (M. Buchin & Wenk, 2020; Moayedi et al., 2019; C. Wang et al., 2021), interaction analysis (e.g., the follower-leader pattern can be abstracted from similar trajectories with a slight delay) (Konzack et al., 2017), clustering and abnormality detection (Djenouri et al., 2021; Mao et al., 2021), trajectory classification (Endo et al., 2016), behavioral mode detection (Bashir et al., 2007; Xiao et al., 2017), and movement prediction (Cheng et al., 2021; D. Choi et al., 2020). However, existing literature lacks a clear systematic review of how movement similarity is used explicitly as distance metrics and implicitly as the background knowledge to complete other movement-related tasks, e.g., movement prediction.

In this thesis, movement similarity is classified into two categories. The first category is *explicit movement similarity*, which is the universally used category. It is defined as the closeness of trajectories in space and time. Explicit movement similarity can be directly quantified using a wide variety of similarity measures, such as Fréchet distance and dynamic time warping. The core of explicit movement similarity are similarity measures, i.e., distance functions. The rudimentary Euclidian distance is perhaps the simplest way to measure how similar two trajectories are. That, however, is often far from practical in reality, due to the lack of temporal information in consideration. Therefore, more suitable similarity measures are adopted from other fields, e.g., dynamic time warping (DTW) (Cleasby et al., 2019; Yingmin Li et al., 2010), which is originally proposed to deal with time series (Berndt & Clifford, 1994), and edit distance (Zhu et al., 2021), which is initially used to compare string similarity (Levenshtein, 1966). Moreover, the past several decades have seen many improved similarity measures that are proposed dedicatedly aiming at movement and trajectory analysis. Dodge et al., (2012) proposed a modified version of edit distance named normalized weighted edit distance (NWED) to first segment a trajectory based on some specific movement parameters (e.g., speed). Accordingly, a trajectory is converted into a string sequence, which is then used for similarity measuring. Sharif & Alesheikh (2017) push the similarity measures closer to being practical in reality by including context-awareness, as movements are, to a great extent, influenced by internal and external contexts.

On the other hand, the second category, *implicit movement similarity*, is defined as the consistency in movement and behavioral patterns of an individual or a group of moving entities. Although it is usually challenging for humans to directly interpret or detect implicit movement similarity, it can serve as the foundation to solve many types of problems. Some of these problems, such as trajectory outlier detection (Belhadi et al., 2021) and transportation mode inference (Dabiri & Heaslip, 2018; Nawaz et al., 2020), may still be solvable using traditional explicit similarity. However, it is the implicit similarity learned by models that make the solutions to other problems possible and effective. These examples include urban vehicle trajectory generation (S. Choi et al., 2021) and trajectory prediction (D. Choi et al., 2020). Implicit movement similarity is learned from historical tracking data mainly using machine learning approaches. Classic machine learning methods, such as the decision tree (Quinlan, 1986) and random forest (Ho, 1995), remain effective in movement analysis (Xiao et al., 2017; Zhang et al., 2020) in the present day, especially when the training data size is relatively small and the movement patterns are not complex. However, with the massive volume of data in the big-data era comes a rising demand for data-driven models to learn more complicated patterns, interactions, and relationships. This demand requires models to have more complex architecture. That is one of the primary reasons why recent years have seen an increasing number of approaches based on neural networks (Jäger, 2019; Yali Li et al., 2019; Mehri et al., 2021). Neural networks have better generalization capacity in many cases and have more flexible architecture, such as convolutional and recurrent neural networks (Lan et al., 2020).

This thesis seeks to first review movement similarity and then apply methodologies based on that to solve problems regarding change detection and trajectory interpolation. The following chapter introduces the background and reviews literature related. Chapter 3 introduces the turkey vulture tracking dataset used in this thesis. After that, Chapter 4 presents a hieratical clustering framework based on explicit movement similarity to detect variations in animal movement and infer their drivers. On the foundation of implicit movement similarity, Chapter 5 proposes a trajectory interpolation method based on a recurrent neural network. Both proposed methods are experimented on the real-world tracking dataset. Finally, the thesis ends with discussions and conclusions.

Chapter 2

Background

This chapter first provides a brief introduction of the five similarity measures, including Fréchet distance, Hausdorff distance, dynamic time warping, longest common subsequence, and normalized weighted edit distance. This is followed by a comparison of different similarity measures in real application scenarios. Then, the background of movement path variation detection with the application of explicit movement similarity is given. Finally, Section 2.3 reviews trajectory interpolation and how implicit movement similarity can be incorporated.

2.1 Movement similarity measures

A trajectory is an ordered sequence of tracking points recorded by a location-aware technology, such as GPS devices, RFID tags, geo-sensors. Each tracking point is captured with information including the location, timestamp, and sometimes other attributes as in Equation (1) (Aghabozorgi et al., 2015; Laube, 2014).

$$p_i = (loc_i, t_i, A_i \dots) \tag{1}$$

To quantify the similarity of trajectories, a variety of distance metrics have been studied. Among them, Fréchet distance, dynamic time warping (DTW), Hausdorff distance, and longest common subsequence (LCSS) are some of the basic metrics (Moayedi et al., 2019; H. Wang et al., 2013). All these four commonly-used distance metrics evaluate trajectory similarity in terms of their geometric shapes by computing point-wise distance between two trajectories. From another perspective, the derived movement attributes, such as speed and acceleration, can also be important in characterizing similar trajectories. Therefore, the normalized weighted edit distance (NWED) computed based on speed patterns is also considered here (Dodge et al., 2012).

2.1.1 Fréchet distance

Fréchet distance considers both the location and the order of the tracking points along two trajectories when measuring their similarity. It is often depicted vividly as the person-dog metric (K. Buchin et al., 2016; Gudmundsson & Valladares, 2015). As illustrated in Figure 2, imagine a person walking on trajectory T_i and a dog walking on T_j, the Fréchet distance is the length of the shortest leash that ensures the connection from the beginning to the end. Note that neither the person nor the dog can walk backwards. Put formally, the Fréchet distance between two trajectories T_i and T_j in a Euclidean space can be computed by Equation (2-3) (Khoshaein, 2013).

$$dist_{Fréchet}\left(T_i\,, T_j\,\right) = \min\left(\|c\|, \qquad c\ is\ coupling\ between\ \ T_i \text{ and } T_j\right) \tag{2}$$

$$\|c\| = max_{k=1}^{K} euc\left(p_i^k, p_j^k\right) \tag{3}$$

where T_i and T_j are two trajectories of length m and n. $K = min(m, n)$. p_i^k and p_j^k are the k^{th} tracking point of trajectory T_i and T_j, respectively. $euc\left(p_i^k, p_j^k\right)$ is the Euclidean distance between point p_i^k and p_j^k.

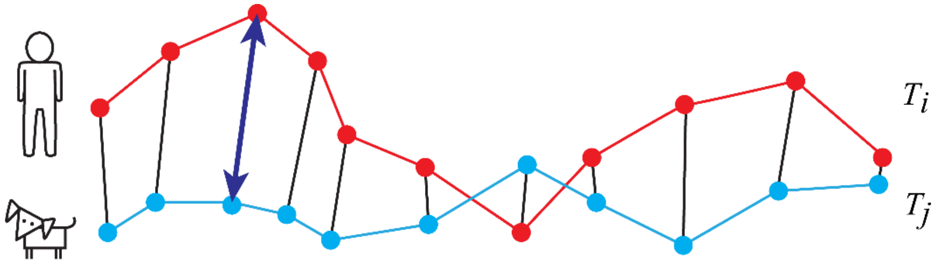


Figure 2. Illustration of Fréchet distance between two trajectories (shown in blue and red) (the largest pair-wise distance, shown in dark blue with arrows)

2.1.2 Hausdorff distance

Hausdorff distance is a metric for evaluating the similarity of the shapes of two trajectories or a subset of them without considering their direction (i.e., the temporal order of tracking points).

Hausdorff distance between two trajectories T_i and T_j selects the maximum unidirectional distance from T_i to T_j and from T_j to T_i. Thus, it measures the maximum degree of mismatching between two trajectories. It does not require the input trajectories to have the same length and has low computational complexity (C. Wang et al., 2021). But it is also sensitive to noise in the data (Meng et al., 2019). Hausdorff distance between two trajectories T_i and T_j in a Euclidean space can be computed by Equation (5-7) (Sun et al., 2021).

$$dist_{Hausdorff}(T_i, T_j) = min\left(h(T_i, T_j), h(T_j, T_i)\right) \tag{4}$$

$$h(T_i, T_j) = \max_{\forall p_i \in T_i}\left(\min_{\forall p_j \in T_j}\left(euc\_dist(p_i, p_j)\right)\right) \tag{5}$$

$$h(T_j, T_i) = \max_{\forall p_j \in T_j}\left(\min_{\forall p_i \in T_i}\left(euc\_dist(p_j, p_i)\right)\right) \tag{6}$$

where $euc\_dist(p_i, p_j)$ is the Euclidean distance between point p_i and p_j.

2.1.3 Dynamic time warping (DTW)

Dynamic time warping (DTW) measures similarity based on matching tracking points in trajectories, and the tracking points can be matched despite having different timestamps. To obtain a better match, trajectories are “compressed” or “stretched” non-linearly to measure similarity while allowing for different lengths or sampling rates. However, DTW is sensitive to outliers because every tracking point in a trajectory has to be matched with at least one point in another trajectory (Toohey & Duckham, 2015). Given two trajectories T_i and T_j of lengths m and n, DTW distance between them in a Euclidean space can be computed by Equation (4) (Berndt & Clifford, 1994; Yingmin Li et al., 2010).

$$dist_{DTW}(T_i\,, T_j\,) = \begin{cases} 0, & m = n = 0 \\ \infty, & m = 0 \parallel n = 0 \\ euc(p_i, p_j) + min\begin{pmatrix} dist_{DTW}(R(T_i)\,, \mathrm{R}(T_j)\,) \\ dist_{DTW}(R(T_i)\,, T_j\,) \\ dist_{DTW}\left(T_i\,, \mathrm{R}(T_j)\right) \end{pmatrix}, & otherwise \end{cases} \tag{7}$$

where T_i and T_j are two trajectories of length m and n. p_i and p_j denote the initial tracking point of T_i and T_j, respectively, and $R(T_i)$ and $\mathrm{R}(T_j)$ indicate subsequences of T_i and T_j after removing the initial point, respectively, $euc(p_i, p_j)$ is the Euclidean distance between point p_i and p_j.

2.1.4 Longest common subsequence (LCSS)

Distinct from the point-wise distance metric mentioned above, the longest common subsequence (LCSS) distance is computed by obtaining the longest subsequence that is common in both trajectories. A common subsequence is a spatial alignment of tracking points that occurs in both trajectories given two deviation thresholds in the x-direction and y-direction. Therefore, it can still be effective when applied to trajectory data of low quality, which is usually due to factors such as device's accuracy limitation and signal interference. For two trajectories T_i and T_j of lengths m and n, LCSS distance is computed by Equation (8) (Rick, 2000).

$$dist_{LCSS}(T_i\,, T_j\,) = \begin{cases} 0, & m = n = 0 \\ 1 + LCSS_{\sigma,\varepsilon}\left(dist_{LCSS}(R(T_i)\,, \mathrm{R}(T_j)\,)\right), & |p_i^x - p_j^x| \le \sigma, |p_i^y - p_j^y| \le \varepsilon \\ max\begin{pmatrix} dist_{LCSS}(R(T_i)\,, T_j\,) \\ dist_{LCSS}\left(T_i\,, \mathrm{R}(T_j)\right) \end{pmatrix}, & otherwise \end{cases} \tag{8}$$

where σ and ε are the user-defined deviation thresholds in the x-direction and y-direction, respectively. The performance of this metric depends heavily on the choice of the two user-defined threshold parameters, and it is a challenging problem to figure out the optimal parameters (Meng et al., 2019; H. Wang et al., 2013).

2.1.5 Normalized weighted edit distance (NWED)

The four similarity measures mentioned above all tackle the task of measuring trajectory similarity from the perspective of trajectories' spatiotemporal geometry. Normalized weighted edit distance (NWED) introduced in Dodge et al. (2012), however, measures similarity from the behavior perspective, which is manifested by movement parameters (MPs), such as speed and acceleration. Given two trajectories T_i and T_j, the selected MP along the trajectory is first transferred into a sequence of symbolic representation based on the deviation from the mean value and sinuosity of the MP time series. Each symbolic representation denotes a MP class, which represents the frequency and amplitude of change of the behavior. That is, each class encodes, for example, how much the speed value deviates from the average speed, and how variable the speed profile is (e.g., high frequency variations versus low frequency variations). In this way, NWED is capable of quantifying the similarity of two trajectories from the perspective of MPs instead of geographic distance. That is its major difference from the four aforementioned spatial distance metrics. However, from another perspective, NWED considers only the MP but not the spatial path of the trajectories. Given P and Q denoting the MP class sequence of T_i and T_j, NWED between the two trajectories of lengths m and n is computed by Equation (9-10) (Dodge et al., 2012).

$$C_{i,j} = \begin{cases} j, & i = 0 \\ i, & j = 0 \\ C_{i-1,j-1}, & i,j > 0 \text{ } and \text{ } P_i = Q_j \\ CostMatrix\left(P_i, Q_j\right) + min\left(C_{i-1,j-1}, C_{i-1,j}, C_{i,j-1}\right), & others \end{cases} \tag{9}$$

$$dist_{NWED}\left(T_i\ , T_j\ \right) = \frac{2 \times C_{m,n}}{m + n + C_{m,n}} \tag{10}$$

where $C_{i,j}$ denotes the element at the i^{th} row and j^{th} column of the derived $m \times n$ dissimilarity matrix, and $CostMatrix\left(P_i, Q_j\right)$ denotes the element at the ${P_i}^{th}$ row and ${Q_j}^{th}$ column of the cost matrix defined by Dodge et al., (2012). NWED can be used in conjunction with other aforementioned geometric similarity measures to quantify both spatial and spatiotemporal trajectory similarity of trajectories.

2.1.6 Comparison of different similarity measures in clustering

Quantifying similarity is significant to a variety of trajectory analysis applications, such as clustering (K. Buchin et al., 2010; Vlachos et al., 2002), outlier detection and removal (H. Liu et al., 2019; Mao et al., 2021), and classification and characterization (Bashir et al., 2007; Juarez et al., 2011). This thesis focuses on trajectory clustering since that is the key and fundamental step to our goal of identifying variation in movement behavior and inferring environmental drivers.

In a clustering application scenario, the distance metrics have different strengths and weaknesses when applied as trajectory similarity measures. A distance metric may have the best performance under a certain circumstance but not under another. The divergent characteristics of using various distance metrics include (but are not limited to) the following aspects:

First, one salient difference worth noting is the characteristic of outlier tolerance. In general, distance metrics based on point matching together with the sum of distance, such as Fréchet distance, Hausdorff distance, and DTW, are more susceptible to outliers than others (Su et al., 2020). This is because noise and outliers usually contribute to the maximum pair-wise distance. On the contrary, LCSS, as a binary-threshold-based distance metric, has better capability of handling noise. If the distance between a pair of matching points from two trajectories exceed the predefined threshold, their distance will not influence the LCSS distance between the two trajectories. In addition, NWED utilizes a sliding window to capture the sequential MPs along a trajectory, thus diminishing the local influence of outliers. Therefore, among the distance metrics discussed above, LCSS and NWED are capable of quantifying similarity with the minimal influence of outliers, while others, especially Fréchet distance and DTW, are susceptible to outliers, and thus can be used for outlier and anomaly detection. While outliers are merely extreme values with consistent patterns in the given dataset, anomalies usually refer to data with different patterns, e.g., distinct behaviors of animals or humans (Park, 2019).

Second, in terms of efficiency, Fréchet distance has the highest computational complexity of $O(nm\log(nm))$, in which n and m denote the lengths of the two trajectories to compare, while that for all other dynamic programming-based distance metrics is $O(nm)$ (Tao et al., 2021). Compared to other metrics, the original Fréchet distance is not recommended for dealing with large quantity of data or real-time analysis. But recent efforts have been made to improve its computational efficiency (Bringmann et al., 2019).

Third, the performance of some distance metrics depends heavily on the parameter setting (e.g., deviation thresholds σ and ε for LCSS) while others, such as DTW, Fréchet, and Hausdorff distance, are parameter-free. Thus, when comparing trajectories of different categories of moving objects (e.g., when applied with the goal of classification and identification), parameter-dependent metrics need to be tuned with regard to the application scenarios, while parameter-free ones can be applied directly.

2.2 Movement path variation detection and environmental driver inference

Intentional movement, defined as the change of spatial locations of individuals over time, is fundamental to ecological and human systems. As movement is arguably the most significant way by which animals and humans respond to changes in their surrounding environment, it can serve as an instrument for environmental response and understanding movement can elucidate the relations between environmental drivers of behavior and demography (Eikelboom et al., 2020; Nathan et al., 2008). In movement ecology, for example, the recent development and popularization of satellite tracking with significant increase in positioning accuracy, e.g., solar-powered GPS collars, have enabled researchers to track the movement of animals over large distances and long periods. With the emerging data-driven movement ecology paradigm, high-frequency animal movement data provide us with new and effective ways of revealing behavioral changes and environmental response through movement patterns.

The abundant animal movement data offer opportunities for new and effective ways of understanding the internal and external factors affecting animal movement and, one step further, their behavioral ecology. Nearly all animals show some variation in movement patterns induced by the environment (Kranstauber, 2019). The variation in animal movement, however, have not been fully explored in past studies to infer the environmental influence (Eikelboom et al., 2020). In previous studies, inferring environmental drivers is often accomplished through relating movement attributes, such as animal activity space and speed and turning angles, to a set of environmental variables (Dodge et al., 2014; Eikelboom et al., 2020; Patterson et al., 2009). In these studies, movement is described through multiple perspectives including geometry (movement paths) and dynamics (movement attributes). Although these multi-dimensional attributes offer a way to capture more information, they also add to the complexity of the study, especially with regards to long-term tracking data of animals representing multi-seasonal migratory behaviors, and they require more computational resources as well.

On the other hand, movement path variations can be detected utilizing spatial and spatiotemporal similarity measures. Similarity analysis can help to aggregate subsets of tracking data representing similar movement behaviors, and hence, it can reduce the complexity of large datasets by breaking them down into simpler structures that may better reveal important patterns captured in the data. For instance, Cleasby et al. (2019) applied five commonly-used similarity measures individually to compare animal movement trajectories, and meanwhile, to evaluate the performance of those measures.

This thesis explores an alternative approach for studying movement path variation. Chapter 4 of this thesis aims to propose an integrated trajectory-similarity-based hierarchical clustering framework that combines different similarity measures. The clustering results are then used to infer the environmental correlates of animal's migration path choices.

2.3 Trajectory interpolation

Tracking devices sample the location of moving entities with mostly regular intervals as sequences of timestamped locations, named trajectories. However, many factors (e.g., battery outage, signal loss) may lead to the interruption of the tracking data recording, which results in missing data (points) in a tracking dataset. A series of consecutive missing tracking points is termed as a *gap* in this thesis. A gap in tracking data may originate from intentional or unintentional artificial factors. Many GPS-enabled collars attached to animals, for example, turn off tracking devices during the night, since they are solar-powered and GPS sensors consume quite a lot of power (Hirakawa et al., 2018). But many animals do not stop moving immediately after the sun goes. This temporal inconsistency leads to gaps in the tracking datasets. However, there is a special case where sometimes sensors shut down to save battery when the animal is not moving. The gap here is only temporal but not spatial, as the animal remains in the same location. This kind of gaps are usually harmless and can be easily detected and differentiated from other ordinary gaps. In terms of gaps in human tracking data, a GPS-included travel survey in three Scottish towns shows that a non-negligible part of participants return with unusable tracking data due to data missing (Siła-Nowicka et al., 2016). On the other hand, the gaps or missing data in trajectories may result from non-manufactured and incidental factors, such as tracking device malfunction and signal communication failures. Gaps originated this way often create a more significant problem since they may appear stochastically. Hence, they are the primary targets that need to be dealt with.

Problems can arise when trajectories containing gaps are utilized in further movement analysis. For example, gaps might influence the calculation of animal home range and daily movement patterns (Wentz et al., 2003). And in both animal and human movement analysis, studying joint movement patterns can be negatively impacted by gaps in trajectories (Benkert et al., 2008). It is worth noting that the problems of gaps cannot be alleviated by increasing sampling rates or positioning accuracy, as they arise from innate reasons of tracking devices, entities being tracked, and signal transmission.

There are in general two types of approaches to deal with gaps in trajectories. The first type is passively throwing away trajectories containing gaps or dividing the trajectories where the time interval between two consecutive tracking points exceeds a predefined threshold. This is straightforward and computationally simple, but it is at the risk of decreasing data volume and even potentially removing implicit long-term patterns if a complete trajectory is divided. The second type of approaches, on the contrary, actively filling in the gap by estimating the locations and other attributes of the missing tracking points from the valid observed ones along a trajectory. This process is termed *trajectory interpolation* in this thesis. It can be viewed as a special case of trajectory prediction, since in both interpolation and prediction, the goal is to estimate the unknown tracking points according to those valid observed ones. Nonetheless, prediction in many cases refers to extrapolation, meaning that observed tracking points only exist on one end, while in an interpolation task, observed tracking points exist on both ends. Trajectory interpolation not only preserves or even increases the volume of a dataset but also preserves the patterns hidden in a complete trajectory, both of which might be crucial for further analysis. Therefore, trajectory interpolation remains a significant problem in both movement ecology and human mobility demanding further attention and study to better utilize real-world tracking data in computational movement analysis (Parent et al., 2013; Ren et al., 2021; Zheng et al., 2014).

The classic and most straightforward trajectory interpolation method is linear interpolation. This method assumes that the entity is moving at a constant speed and heading in the interpolated area, which is not realistic since it underestimates the complexity (tortuosity) of real trajectories (Rowcliffe et al., 2012). The real trajectory is often not a straight line segment between the origin and destination or between two tracking points where interpolation is needed. One important reason behind it is that some extent of stochasticity exists in movement. Thus, some researchers model movement as a probabilistic random process, e.g., random walks (Rowcliffe et al., 2012; Technitis et al., 2015), or with uncertainty—a potential path area (PPA) from the field of time geography (Ahearn et al., 2017; Hägerstrand, 1970; Miller, 2005), for example. Linear interpolation can be considered a special circumstance of a random walk model. It represents the shortest and most-likely path between the start and end points if no other information, especially no other observed tracking points, are taken into account (Winter & Yin, 2010). In general, random walk interpolation models have the capability to capture and utilize random movement patterns, thus successfully modeling the stochasticity of movement (Technitis et al., 2015; Wentz et al., 2003). Nevertheless, one major challenge is precisely parameterizing the probabilistic rules, which is no easy task and often requires sufficient domain knowledge of the moving entity. Alternatively, the knowledge needed to interpolate a gap can be unraveled from the valid observed tracking data, especially the very trajectory that contains the gap. In view of trajectories' geometrical characteristics, researchers have explored various interpolation methods based on curve fitting, such as cubic splines, Bézier curves, and polynomial curves (Kolendo & Śmierzchalski, 2016; Tremblay et al., 2006; Yu & Kim, 2006). The principle behind these approaches is that a trajectory is viewed as a curve whose shape can be represented mathematically. However, they focus on the spatial features only but neglect the temporal or spatiotemporal features, such as time and speed, implied in trajectories. To incorporate spatiotemporal features, Long (2016) and S. Guo et al. (2021) put forward interpolation methods based on kinematic information extracted from the observed tracking points. But this kind of interpolation methods are sensitive to sampling rate and the speed of moving entities. In general, they need tracking datasets of fast-moving entities with high sampling rates to achieve a satisfactory performance. Therefore, there is a need to design interpolation methods with broader applicability and better robustness.

Recent years have seen the versatility and effectiveness of machine learning methodologies proven in an increasing number of domains and fields. In movement analysis, machine learning approaches have been used to predict animal behavior states (Michelot et al., 2016; Patterson et al., 2008), simulate animal personality related to movement preferences (Spiegel et al., 2017), extract road network information by modeling vehicle trajectories (Yang et al., 2022), and classify transportation modes (Nawaz et al., 2020). As for trajectory prediction, Gupta et al. (2018) and Sadeghian et al. (2019) utilize generative adversarial networks (GAN) to predict physically and socially acceptable pedestrian trajectories according to pixel-based pedestrian tracking datasets captured by cameras. In terms of tracking data recorded by GPS-enabled devices, many researchers tend to use recurrent neural networks (RNNs), the improved variations of the classic RNNs in particular, long short-term memory (LSTM) and gated recurrent unit (GRU) (Cho et al., 2014; Hochreiter & Urgen Schmidhuber, 1997). RNNs are naturally suitable for dealing with sequential data, especially time series, to which category trajectories happen to belong. The classic RNNs suffer from problems of vanishing and exploding gradients, but these problems are successfully overcome with LSTM and GRU by introducing gates in the recurrent unit to control the information learned from the input, stored in memory, and output by the model. Qin et al. (2021) proposes a trajectory prediction model based on LSTM and Kalman filter. The outputs of LSTM are first restricted to a reasonable range and then refined by Kalman filter to produce the final prediction results. In terms of trajectory interpolation in particular, Rew et al. (2019) use a random forest method to interpolate the missing tracking points one at a time along a trajectory. The applicability and accuracy of that method are limited, since it neglects the implicit movement patterns contained in trajectories that need to be extracted sequentially. Considering all trajectories as a whole, Hirakawa et al. (2018) learns a reward (preference) map through inverse reinforcement learning (IRL) from the observed tracking data. Then, a trajectory containing a gap can be reconstructed to a complete one according to the reward map. However, with the same pair of start and end points to interpolate, their method always produces the interpolation result. Because the individuality is not considered when the whole tracking dataset is used to learn a highly integrated reward map.

To address that, Chapter 5 of this thesis proposes a GRU-based trajectory interpolation model with the encoder-decoder architecture to learn both joint and individual movement patterns. It learns joint movement patterns as it is trained on historical tracking data composed of multiple trajectories of different individuals. When interpolating the gap, the model first encodes that very trajectory containing the gap to learn individual patterns, and then decode the complete trajectory with the gap filled in.

Chapter 3

Turkey vulture tracking dataset

The turkey vulture (*Cathartes aura*) data used in this thesis was originally obtained from Bildstein et al. (2014) and was continuously collected through 2021 (Bildstein et al., 2021; Mallon et al., 2021). The dataset is archived and available on Movebank[1] (Kranstauber et al., 2011), an online platform that help researchers manage, share and analyze animal movement data. Turkey vultures are the most widely distributed, as well as the most abundant obligate avian scavenger in North America (Dodge et al., 2014). Based on the time of year and location, the collected tracking points in the dataset have been segmented to four phases: breeding grounds in Canada, non-breeding (or wintering) grounds in Venezuela, fall migration from Canada to Venezuela, and spring migration from Venezuela to Canada.

The original dataset is collected from 13 turkey vultures with a total time span of approximately 15 years. However, the time span over which each bird has been tracked varies (see Figure 3). The mean tracking time span per individual bird is 876.1 days (2.4 years) with a standard deviation of 1284.7 days (3.5 years). The mean time interval for tracking is 1.8 hours with a standard deviation of 5.2 hours. Since this thesis focuses on movement processes, only tracking points representing the migratory phases, i.e., fall and spring migrations, are used in the analysis. Here, the time interval threshold, $time\_th$, is empirically set to 720 hours (roughly 1 month) to divide the trajectory of a bird into multiple trajectories recording different migration events. Adjacent tracking points with a time interval greater than $time\_th$ should not be connected since

[1] https://www.movebank.org/

the interval is too large to model the trajectory path choice between the two points. The trajectories after preprocessing are visualized in Figure 4.

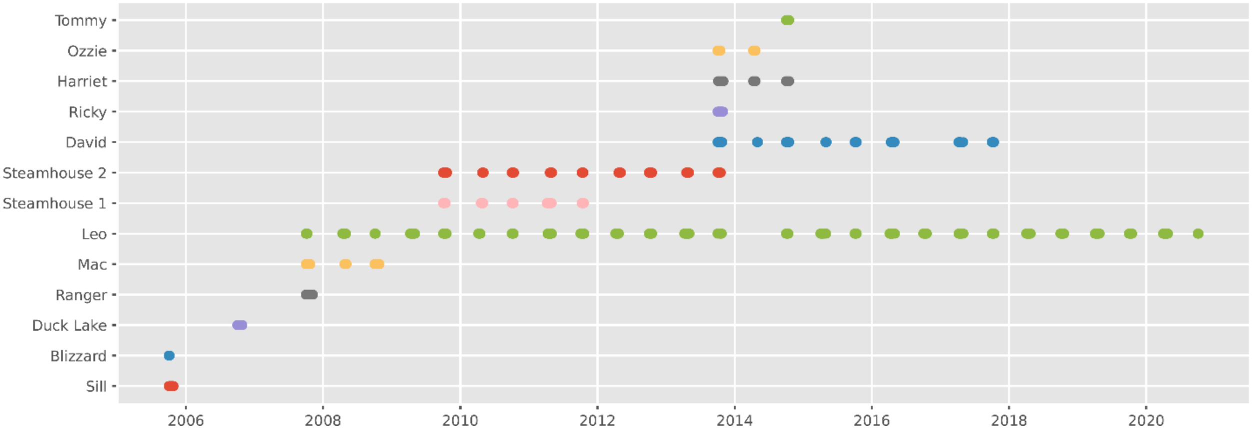


Figure 3. Tracking time span for each turkey vulture after preprocessing. On the y-axis are the turkey vulture names. Colored points represent timestamps at which tracking data were collected.

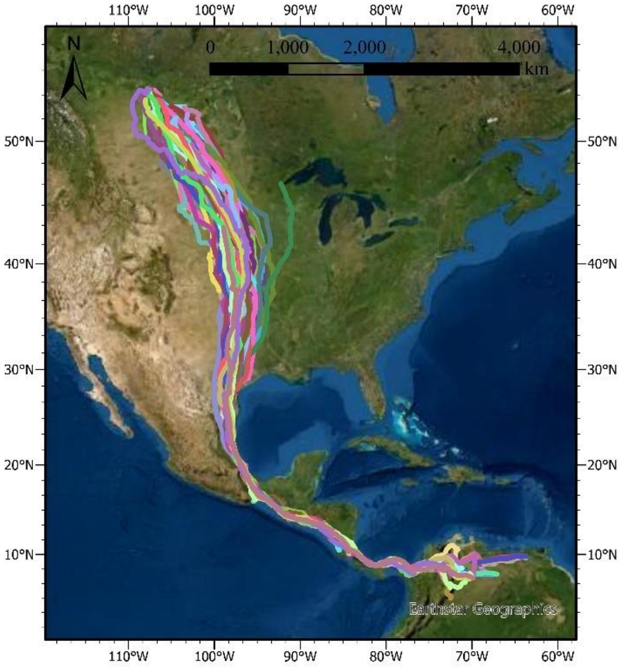


Figure 4. Turkey vulture migration trajectories

Chapter 4

Animal movement path variation detection and driver inference: a movement-similarity-based framework

As movement is arguably the most significant way by which animals respond to changes in their surrounding environment, it is important to detect variations of movement paths based on trajectory similarity and infer the latent environmental drivers. Given the strengths and weaknesses of different trajectory similarity metrics, as stated in Chapter 2, this study proposes an integrated framework that exploits the relative strengths of every distance metric so that distinct clusters can be revealed.

4.1 Methodological framework

Figure 5 illustrates the proposed framework. In this framework, trajectory data are first preprocessed and enriched with environmental variables. Then, trajectory-similarity analysis is performed based on a hierarchical clustering framework that integrates the five aforementioned similarity measures to find the latent clusters of trajectories based on similarity between trajectory shapes and movement parameters. Next, the obtained clusters of trajectories are characterized by environmental variables. Then, the distribution of environmental variables of each cluster is analyzed to infer the ones that might influence the trajectory path choice of animals. This framework is flexible to employ any number of similarity measure using a hierarchical clustering approach. The strength of this framework as compared to only applying one similarity measure is that it is able to produce the most distinct clusters at each hierarchy according to the most salient features captured by different similarity measures.

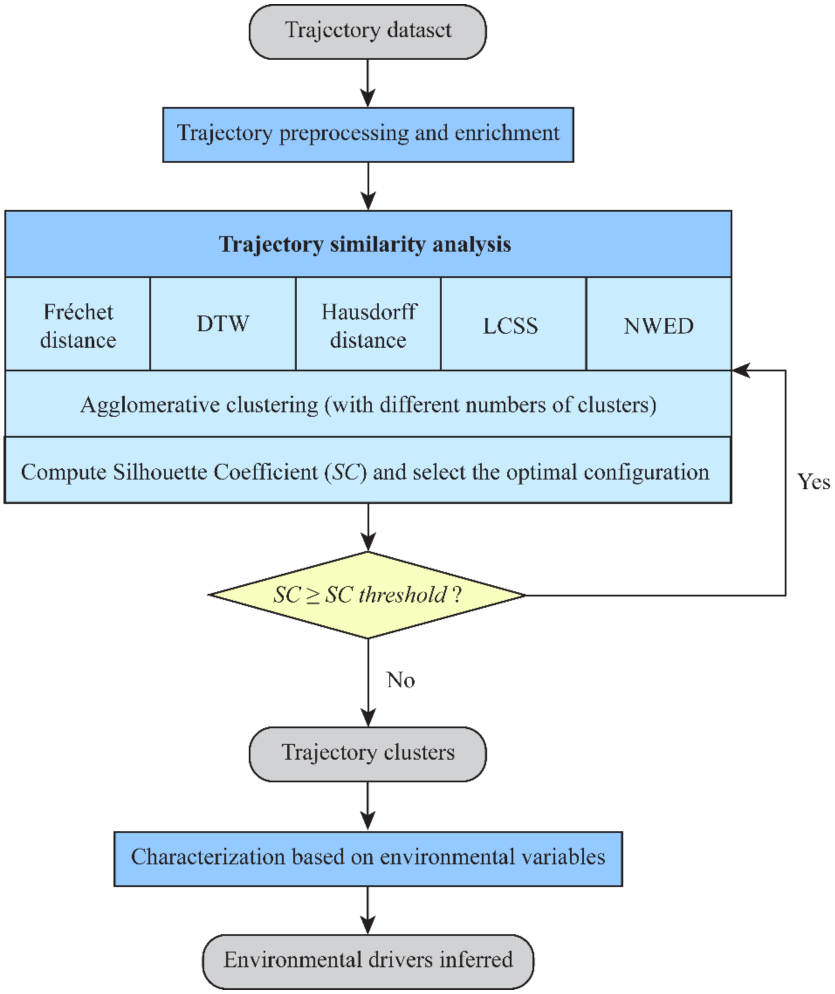

Trajectory dataset
Trajectory preprocessing and enrichment
Trajectory similarity analysis
Fréchet distance
DTW
Hausdorff distance
LCSS
NWED
Agglomerative clustering (with different numbers of clusters)
Compute Silhouette Coefficient (SC) and select the optimal configuration
SC ≥ SC threshold ?
Yes
No
Trajectory clusters
Characterization based on environmental variables
Environmental drivers inferred

Figure 5. Workflow of the hierarchical clustering framework

4.1.1 Trajectory preprocessing and enrichment

For each tracked animal (identified uniquely by different tags or names), the tracking points in its trajectory dataset are connected in chronological order to form a trajectory. This study only focuses on GPS tracking points that represent migration behaviors. Since migration is a periodic activity representing a frequently occurring pattern, the trajectory of each tracked individual is segmented based on the time intervals of the migration season (e.g., fall or spring migration). After preprocessing, each tracking point is enriched with a series of environmental variables identified as potentially effective factors based on domain knowledge. The environmental variables can be obtained through a trajectory enrichment process by interpolating and integrating remote sensing and weather reanalysis models data with movement data, for example, using the Env-DATA service in movebank.org (Brum-Bastos et al., 2016; Dodge et al., 2013).

4.1.2 Trajectory similarity analysis and clustering

After preprocessing and enrichment, trajectory-similarity analysis is performed based on a hierarchical clustering framework to find the latent trajectory clusters. Hierarchical clustering is an approach that derives a series of hierarchies of clusters using agglomerative (bottom-up) or divisive (top-down) algorithms. In this study, the proposed framework takes the form of divisive hierarchical clustering. It starts by creating the "root" of the hierarchical cluster "tree" and continues by dividing the entire dataset into clusters, i.e., subsets of similar objects.

At each hierarchical level of the framework, this study performs clustering by choosing the optimal distance metric together with the optimal number of clusters. In theory, this framework can be applied to any kind of clustering algorithms. However, clustering results might vary with different clustering algorithms. This study applies the agglomerative clustering with average linkage, a widely used algorithm in trajectory clustering (Dodge et al., 2012; Y. Guo et al., 2016; Miller, 2008). Agglomerative clustering is a bottom-up clustering approach that searches for partitions of similar entities. It starts from the initial partition that treats every object as its own cluster. Then, based on a linkage criterion, it partitions data in a way that maximizes the difference between clusters and minimizes the distance within clusters. Average linkage, which quantifies the similarity between two clusters as the average distances between all pairs of observations of the two clusters, is probably the most popular metric to evaluate similarity between clusters in agglomerative clustering (Moseley & Wang, 2017). In a nutshell, herein this study proposes a divisive hierarchical framework, at each level of which an agglomerative clustering is applied.

Since labeled trajectories are often unavailable and the goal of the proposed method is to find latent clusters, external validation indices, such as accuracy and purity, cannot be used.

Therefore, the silhouette coefficient (SC) is used as an internal validation index to evaluate the result of a clustering configuration, i.e., the distance metric together with the number of clusters. SC is a metric that quantifies how similar are objects within each cluster compared to those in other clusters. SC ranges from -1 to 1, where a higher SC indicates a better clustering result. An SC of 1 indicates that clusters are well apart from each other and clearly distinguished, an SC of 0 means that the distance between clusters is not significant, and an SC of -1 suggests that clusters are wrongly assigned. SC can be calculated with any distance metric and is often used to figure out the parameter configuration for a clustering task (F. Liu et al., 2021). The SC of a clustering result can be computed by Equation (11) (F. Liu et al., 2021; Rousseeuw, 1987).

$$SC = \frac{1}{n_c}\sum_{i=1}^{n_c}\frac{b_i - a_i}{max(a_i, b_i)} \tag{11}$$

where n_c is the number of clusters, a_i denotes the mean distance from each object to all other objects in cluster i, and b_i denotes the mean distance from each object to all objects in the next nearest cluster.

In the proposed framework, the optimal clustering configuration is chosen at each hierarchy that leads to the highest SC. This clustering process is recursive, and the recursion continues until the stop criterion has been met. This study sets the stop criterion at each hierarchical level as $SC < SC\_threshold$. Because a low SC indicates that the clustering process at that hierarchical level no longer contributes much to obtaining distinct and meaningful clusters from the trajectory dataset, and thus should be abandoned.

4.1.3 Cluster characterization based on environmental variables

The next step after obtaining the trajectory clusters from the dataset is to interpret the clustering results in order to infer the environmental drivers of animals' migration-trajectory path choice. To test whether an environmental variable makes a significant effect on the trajectory, this study compares the environmental variable distribution of each cluster against the background distribution of that variable. The background distribution is assumed as the null distribution of that variable. The hypothesis is that an environmental variable distribution within a trajectory will not be different than the background if animals choose their migration-trajectory path without any information or preference with regard to that environmental variable (Bohrer et al., 2012). For every trajectory cluster, a Kolmogorov-Smirnov (KS) test is run (implemented using SciPy, a Python package (Virtanen et al., 2020)) on each environmental variable distribution against its background distribution. To prevent pseudo-replication due to the arbitrary high number of trajectory points in high frequency GPS data, this study sets the degree of freedom in the KS tests equal to the number of trajectories in that cluster, and not the number of location points following Bohrer et al. (2012). Finally, a $p-value$ of 5% is used as the significance threshold for the effect of environmental variable.

4.2 Data

4.2.1 Trajectory dataset

As shown in Figure 4, most of the variation among trajectories are manifested in North America (from Saskatchewan, Canada to Texas, USA), whereas all trajectories converge over Central America, and trajectories below 30N (from Texas, USA to Venezuela) are almost identical as the birds seem to follow the narrow available land corridor. Because this study is interested in identifying and characterizing variations in the migration paths of these birds, only the variable portion of the trajectories are focused on. Therefore, the trajectories are truncated at latitude 30N. After the preprocessing and truncation, the turkey vulture dataset consists of 62 trajectories from

13 turkey vultures with 17,118 tracking points (as in Figure 6) and the properties are illustrated in Table 1. The numbers of fall and spring migration trajectories are 37 and 25, respectively.

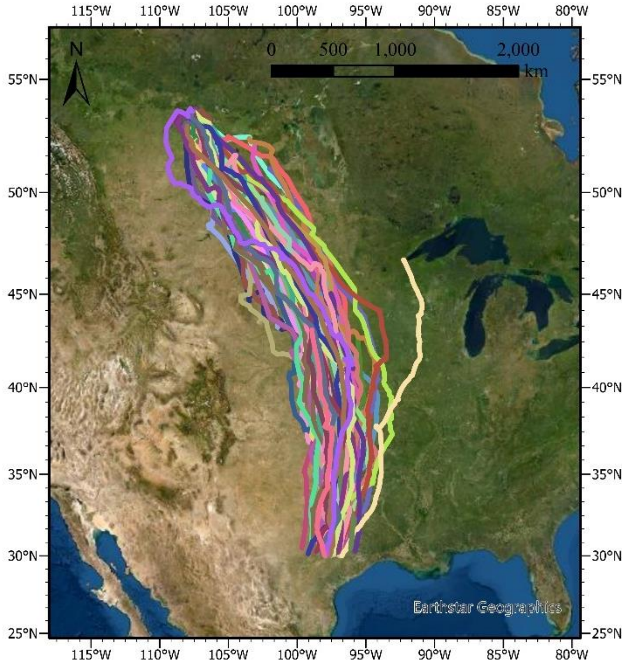


Figure 6. Turkey vulture migration trajectories truncated at latitude 30N. Each trajectory representing a bird during one season is shown by a unique color.

Table 1. Properties of the turkey vulture trajectory dataset after preprocessing (std represents standard deviation).

	Time interval (h)	Speed (km/h)	Trajectory length (km)	Tracking time span per trajectory $(days)$
$mean$	1.6	7.3	2510.5	18.2
std	5.4	15.5	777.1	7.4

4.2.2 Environmental variables

This study selects the environmental variables that were previously identified as effective external factors that impact turkey vulture movements (Bohrer et al., 2012; Dodge et al., 2014). The environmental data are obtained and annotated to trajectories using Movebank's Env-DATA toolpack (Dodge et al., 2013). A brief definition to each environmental variable used in this study is provided as follows:

Movebank Thermal Uplift (from ECMWF) (m/s): Velocity of the rising air as it is heated by solar radiation during the day. It is an Env-DATA-derived variable, calculated using weather data from ECMWF Global Atmospheric Reanalysis based on equations provided in Bohrer et al., (2012).

Movebank Orographic Uplift (from ASTER DEM and ECMWF): Velocity of the rising air formed when sloping terrain forces horizontal winds to higher elevation (Bohrer et al., 2012). It is calculated based on the elevation data from the ASTER ASTGTM2 30-m DEM and weather data from the ECMWF Global Atmospheric Reanalysis.

MODIS Land Vegetation Indices 250m 16d Aqua NDVI: The Normalized difference vegetation index (NDVI) is a measure of live green vegetation based on remote sensing reflectance. Higher values represent more live green vegetation.

ECMWF Interim Full Daily SFC Temperature (2 m above Ground): Air temperature at 2 meters above the earth's surface. It is calculated based on the interpolation between the lowest level of the model and the earth's surface, with atmospheric conditions taken into account.

ECMWF ERA5 SL Total Precipitation: Accumulated amount of precipitation over the past hour. Values represent the depth of the water if it is spread evenly over a grid box.

Tailwind: Velocity of the wind in the direction of travel of a moving entity. It is computed based on ECMWF ERA5 SL Wind (10 m above Ground U Component) and ECMWF ERA5 SL Wind (10 m above Ground V Component), which are east-west component and north-south component of the wind at 10 meters above the earth's surface, respectively.

Crosswind: Velocity of the wind that is perpendicular to the direction of travel of a moving entity. It is computed based on ECMWF ERA5 SL Wind (10 m above Ground U Component) and

ECMWF ERA5 SL Wind (10 m above Ground V Component), which are east-west component and north-south component of the wind at 10 meters above the earth's surface, respectively.

4.3 Results

In this section, first, a comparative evaluation of the performance of various distance metrics on trajectory similarity analysis is provided. Then, using the hierarchical clustering framework proposed above, this study obtains the trajectory clusters and characterize them with environmental variables to infer the ones leading to variation in movement path.

4.3.1 Initial clustering and outlier detection

Pair-wise distances between trajectories are computed using all five distance metrics including Fréchet distance, DTW, Hausdorff distance, LCSS, and NWED based on the speed profile. For each clustering configuration, i.e., a distance metric together with the number of clusters $n\_clusters$ obtained using the silhouette coefficient (SC) to evaluate the clustering performance (see Figure 7). For Fréchet distance results, the SC decreases as $n\_clusters$ increases, which indicates that there is no need to test on larger $n\_clusters$. According to the result shown in Figure 7, the internal validation suggests that 2 is the optimal number of clusters for Fréchet and Hausdorff distance and NWED, and 3 for DTW and LCSS.

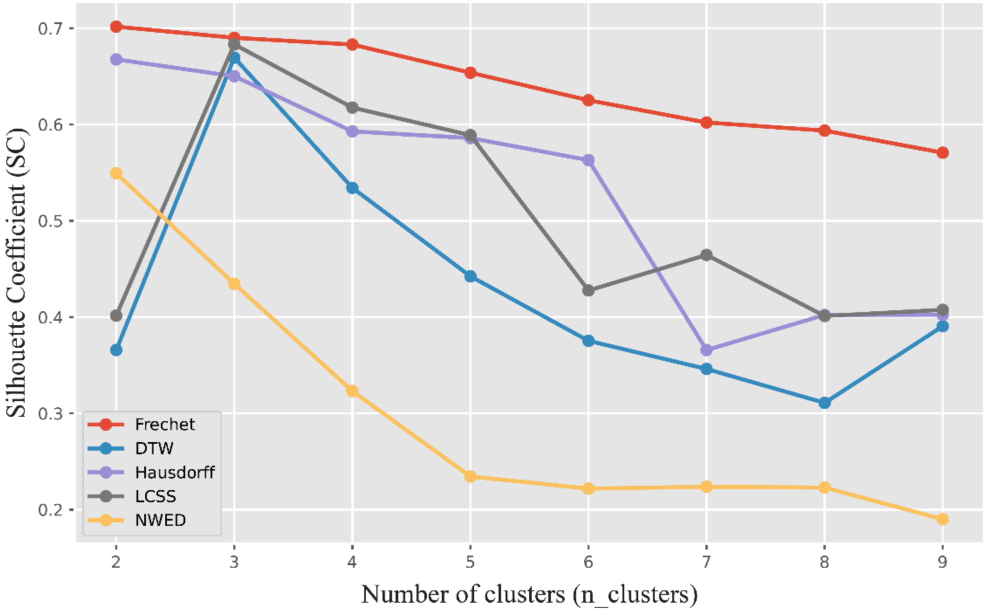


Figure 7. Silhouette coefficient (SC) of initial clustering results using different trajectory similarity measures.

To interpret the initial clustering result, the obtained clusters are characterized using the optimal $n\_clusters$ for each distance metric as an external validation. As shown in Table 2, the properties of each cluster are characterized by the number of tracking points, trajectory length, fall migration counts, and spring migration counts. Fall/Spring migration counts are the number of trajectories in each cluster that depict fall or spring migration trajectories, respectively. It is important to note that a large portion of the tracking points in this dataset, specifically those that are collected before 2013, are already manually labeled with their migration seasons by the data owner and field experts based on the dates and the overall direction of the movement path (Bohrer et al., 2012; Dodge et al., 2014). As mentioned earlier, turkey vultures migrate south in the fall and return north in the spring. Therefore, in general, distance metrics that take direction into account, such as Fréchet distance, DTW, and LCSS, are capable of differentiating the fall migration trajectories from those representing the spring migration as long as the trajectories have similar lengths.

Table 2. Properties of trajectory clusters obtained using the optimal configuration (std represents standard deviation).

Distance metric	Cluster	Number of tracking points per trajectory ($mean\ \pm std$)	Trajectory length (km) ($mean\ \pm std$)	Fall migration count	Spring migration count
Fréchet distance	1	$283\ \pm\ 190$	$2291.5\ \pm 739.3$	37	0
	2	$266\ \pm 137$	2834.5 ± 716.0	0	25
DTW	1	278 ± 186	2399.6 ± 598.6	35	0
	2	266 ± 137	2834.5 ± 716.0	0	25
	3	358 ± 229	400.4 ± 243.1	2	0
Hausdorff distance	1	273 ± 168	2580.8 ± 684.6	35	25
	2	358 ± 229	400.4 ± 243.1	2	0
LCSS	1	278 ± 186	2399.6 ± 598.6	35	0
	2	266 ± 137	2834.5 ± 716.0	0	25
	3	358 ± 229	400.4 ± 243.1	2	0
NWED	1	280 ± 169	2534.8 ± 759.6	36	25
	2	27 ± 0	1026.0 ± 0	1	0

Some trajectories in the dataset are outliers, which mainly consist of incomplete trajectories and those significantly deviate from the general migration corridor. The outliers probably result from tracking anomalies, such as signal loss or interrupted data collection. In the first stage of clustering these outliers can be detected automatically and removed before further analysis. However, if any such anomalous tracks remain in the data, as shown here, the proposed hierarchical clustering framework can be used to identify and remove them automatically. In terms of trajectory outlier detection, DTW, Hausdorff distance, and LCSS agree on two detected outliers (shown in red in Figure 8), while NWED identifies another trajectory as the only outlier (shown in orange in Figure 8). NWED is the most special metric in this study. It focuses on the similarity in the movement parameter (MP) profile derived from the trajectories, the speed profile in this study, while all other metrics focus on the trajectory shape and the spatial location of tracking points. Therefore, when applied to detect outliers, NWED is good at detecting outliers that manifest conspicuously different patterns in the MPs selected, which is especially helpful if those patterns are not well revealed in spatial locations or the shape of trajectories. However, NWED performs poorly if the goal is to detect trajectory outliers that deviate spatially from the mainstream. Fréchet distance, on the other hand, fails to detect any outliers from the dataset when choosing the $n\_clusters$ with the highest SC. That is probably because Fréchet distance mainly considers the maximum distance between matched tracking points without considering the length of trajectories. Interestingly, none of the measure identifies the single trajectory that deviates to the east (trajectory ID: 48) as an outlier. That is because that the difference between that trajectory and the mainstream is much smaller than the difference between the outliers detected and the mainstream.

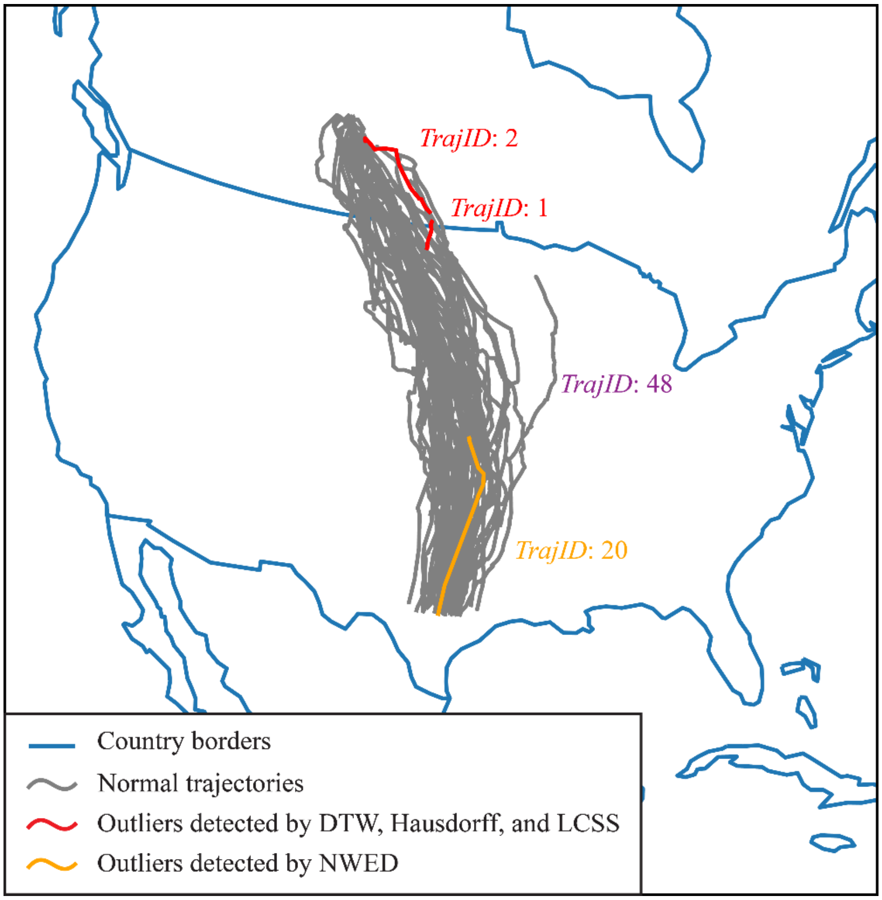
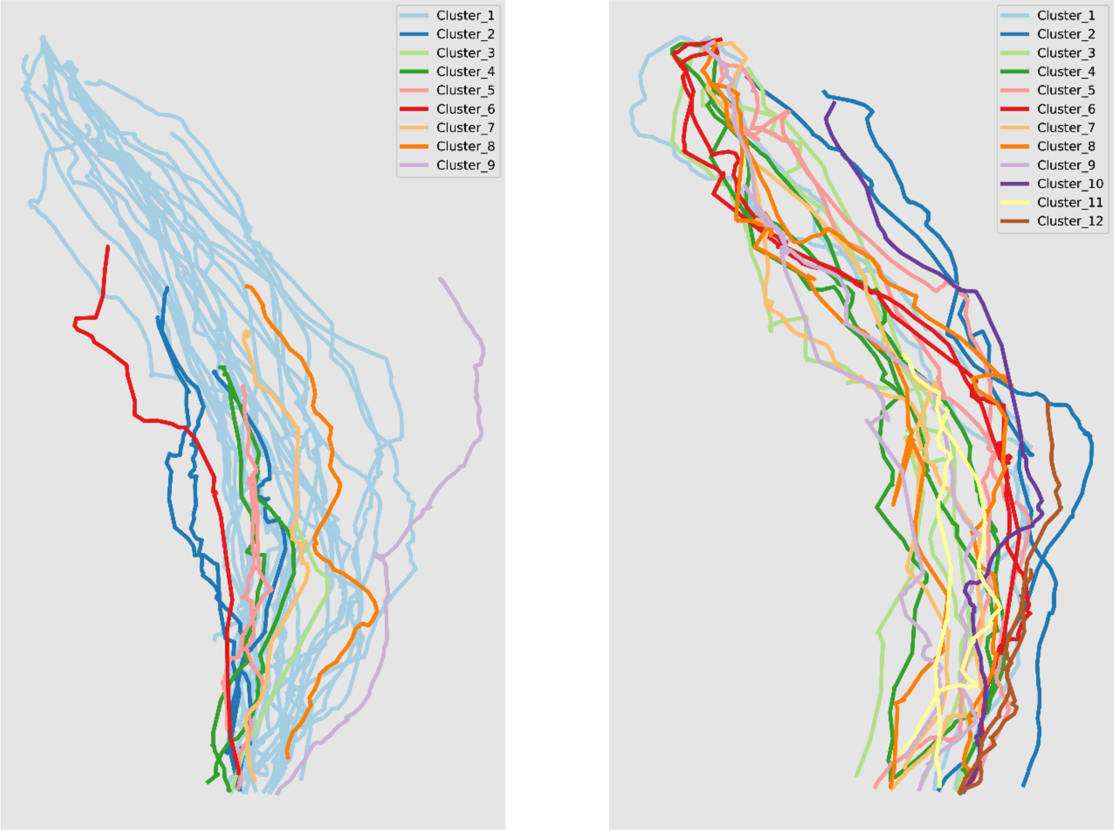


Figure 8. Outliers detected by DTW, Hausdorff distance, and LCSS (in red), and by NWED (in orange). $TrajID$ represents trajectory ID.

4.3.2 Hierarchical clustering results

Based on the initial clustering results, the outliers detected in the first clustering stage are removed before further hierarchical clustering as these outliers do not represent a complete migration behavior. Hierarchical clustering is performed on the first two clusters of Fréchet distance, DTW and LCSS, and the first cluster of Hausdorff distance and NWED, respectively. In this study, the stop criterion is set to $SC\_th = 0.5$ since clustering results with SCs above that value are often considered satisfactory (Agarwalla & Minz, 2018; Dos Santos et al., 2011; Eler et al., 2015), and moreover, after multiple experimentation on this dataset, it was found that $SC\_th = 0.5$ is capable of capturing the inherent characteristics of each cluster without dividing the trajectories into too fine clusters (the extreme scenario is only one trajectory in each cluster). With the proposed hierarchical clustering methodology, even though initial clustering results vary among different metrics, almost the same final clustering result can be reached following similar sequences of clustering configurations at each hierarchical level (see Appendix for detailed hierarchical clustering results). The outliers with trajectory ID 1 and 2 (shown in red in Figure 8), which have not been detected by Fréchet distance at the initial clustering step, are successfully detected and separated from the mainstream through the hierarchical clustering. To achieve that, based on the SC computed, the methodology selects DTW with 2 as $n\_clusters$ at the first hierarchy. Then, for the following hierarchies, the result starting from Fréchet distance follows the same sequence of clustering configurations as DTW and LCSS. Results starting from Hausdorff distance and NWED are two special cases. Since Hausdorff distance is not direction-sensitive, its initial clustering result can only be used to remove outliers but cannot differentiate fall migration trajectories from spring ones. At the next hierarchical level, fall and spring migration trajectories are separated, as Fréchet distance together with $n\_clusters = 2$ is selected. Finally, it reaches the same final result as DTW and LCSS. The NWED result is the most interesting one. Even though the two outliers (with trajectory ID 1 and 2, shown in red in Figure 8) are detected and separated at the second hierarchy, the trajectory detected and removed as an outlier by NWED in the initial step (with trajectory ID 20, shown in orange in Figure 8) affects the final result of hierarchical clustering. That is probably because the numbers of trajectories in the final clusters are usually small, and thus missing one trajectory significantly affects the similarity between one cluster and another.

In this study, the common hierarchical clustering result shared by Fréchet distance (after removing the outliers detected at the first hierarchy), DsTW, Hausdorff distance and LCSS is adopted as the final clustering result to perform further analysis and interpretation. Figure 9 visualizes the trajectory clusters obtained. Interestingly, for fall migration (southbound), most trajectories fall into a major cluster (cluster 1). Since the stopping criterion is based on the SC threshold, it means that none of the distance metrics applied in this study are able to find distinct sub-clusters for cluster 1. This indicates that when the birds migrate south in the fall, turkey vultures usually follow similar spatial paths. For the northbound spring migration, however, distinct clusters have been identified, which manifests more variability in their migration paths (see Figure 9b).



(a) Fall migration

(b) Spring migration

Figure 9. Final trajectory clusters obtained from the hierarchical clustering (common result shared by Fréchet distance (after removing the outliers detected at the first hierarchy), DTW, Hausdorff distance and LCSS)

Next, the hierarchical clustering result is characterized and interpreted with trajectory properties. After the outliers are removed, there are 35 fall migration trajectories and 25 spring migration ones. The detailed information of each cluster of fall and spring migration trajectories is presented in Table 3 and Table 4, respectively, and the boxplot of trajectory lengths for the clusters is presented in Figure 10. The trajectory length seems to be a determining factor in clustering, especially for the fall clusters. Turkey vultures named *Leo*, *Steamhouse1* and *Steamhouse2* share similar trajectory paths, possibly indicating that they share similar movement patterns in terms of their path choice.

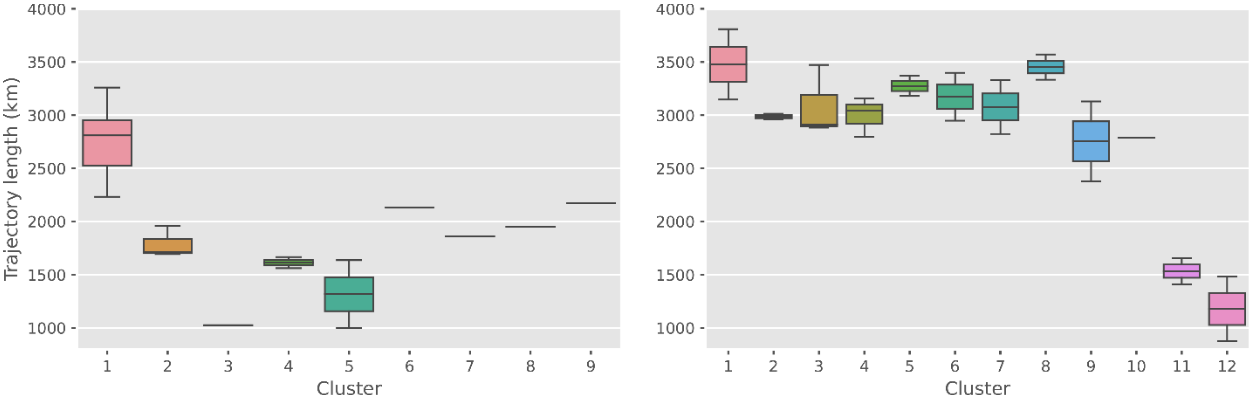
Table 3. Trajectory properties for the fall migration clusters (std represents standard deviation).

Cluster	Trajectory ID	Individual name	Migration year	Trajectory length (km)
1	0	*Sill*	2005	2368.4
	3	*Ranger*	2007	3015.4
	4	*Mac*	2007	2666.7
	11	*Leo*	2009	2893.9
	15	*Leo*	2011	2948.2
	17	*Leo*	2012	2871.2
	19	*Leo*	2013	2788.5
	24	*Steamhouse 1*	2011	2540.6
	25	*Steamhouse 2*	2009	2674.1
	29	*Steamhouse 2*	2011	2354.2
	31	*Steamhouse 2*	2012	2511.4
	33	*Steamhouse 2*	2013	2232.5
	34	*David*	2013	3083.8
	36	*David*	2014	2626.4
	42	*Ricky*	2013	3259.5
	43	*Harriet*	2013	3241.6
	45	*Harriet*	2014	3149.2
	46	*Ozzie*	2013	2346.1
	49	*Leo*	2014	2941.3
	51	*Leo*	2015	2423.2
	53	*Leo*	2016	2810.2
	55	*Leo*	2017	2925.4
	59	*Leo*	2019	2954.1
2	7	*Leo*	2007	1693.4
	13	*Leo*	2010	1958.3
	22	*Steamhouse 1*	2010	1711.6

3	20	*Steamhouse 1*	2009	1026.0
4	57	*Leo*	2018	1666.2
	61	*Leo*	2020	1561.6
5	6	*Mac*	2008	1636.2
	9	*Leo*	2008	998.0
6	27	*Steamhouse 2*	2010	2134.5
7	38	*David*	2015	1859.6
8	41	*David*	2017	1952.6
9	48	*Tommy*	2014	2170.7

Table 4. Trajectory properties for the spring migration clusters (std represents standard deviation).

Cluster	Trajectory ID	Individual name	Migration year	Trajectory length (km)
1	54	*Leo*	2017	3146.5
	60	*Leo*	2020	3809.6
2	40	*David*	2017	2957.5
	47	*Ozzie*	2014	3013.6
3	10	*Leo*	2009	3473.1
	21	*Steamhouse 1*	2010	2882.7
	44	*Harriet*	2014	2909.5
4	8	*Leo*	2008	3158.3
	12	*Leo*	2010	3040.8
	30	*Steamhouse 2*	2012	2796.2
5	50	*Leo*	2015	3369.3
	56	*Leo*	2018	3179.5
6	18	*Leo*	2013	2948.2
	52	*Leo*	2016	3399.1
7	14	*Leo*	2011	3330.7
	23	*Steamhouse 1*	2011	2824.3
8	16	*Leo*	2012	3333.4
	32	*Steamhouse 2*	2013	3571.3
9	28	*Steamhouse 2*	2011	2378.3
	58	*Leo*	2019	3128.9
10	39	*David*	2016	2789.2
11	5	*Mac*	2008	1656.9
	26	*Steamhouse 2*	2010	1409.5
12	35	*David*	2014	876.6
	37	*David*	2015	1480.4



(a) Length of fall migration

(b) Length of spring migration

Figure 10. Boxplot of trajectory lengths for all clusters

4.3.3 Clustering result characterization and interpretation

To explore the potential relationship between environmental variables and the paths choice of turkey vultures, the distributions of environmental variables of each trajectory cluster (i.e., environmental conditions observed at tracking locations of the turkey vultures, hereafter referred to as ‘used conditions’) and those of the background (i.e., environmental conditions available in the general study area, hereafter referred to as ‘available conditions’) are compared. This study randomly sample the background, representing an area and time period for the ‘available condition’, in a bounding box with the most peripheral tracking points (north, south, west, east) as the boundary during the tracking duration (Bohrer et al., 2012). This bounding box is used to compute the distribution of ‘available conditions’ for each environmental variable. Even though the total time span of the trajectory dataset is approximately 15 years, the tracking points recording migrations only cover a smaller proportion of the data (see Figure 3). That is why when generating background samples for the ‘available conditions’, this study only randomly selects from the time range where there are tracking data (marked with colors in Figure 3), and more importantly, each cluster is only compared with the background sample from the same years and same migration season (fall or spring) in which the trajectories in that cluster were recorded. The reason is because environmental conditions in different years and different seasons may vary. Since turkey vultures do not migrate at night (Bohrer et al., 2012), only daytime timestamps were used for background sampling. The total number of random samples selected is 17118, the same number of preprocessed tracking points. These points are then annotated with environmental variables such as thermal uplift, NDVI, and temperature, using the Env-Data system. Most of these variables are directly derived from remote sensing data. However, tailwind and crosswind are special as they are computed based on movement headings instead. To compute these variables at the selected points, the distribution of turkey vulture headings of fall and spring migrations is modeled, respectively. That is, at each sample point a random heading is taken from the distribution of fall and spring migration to compute tailwind and crosswind of the background.

The Kolmogorov-Smirnov (KS) test is used to test the distribution difference of environmental variables between each cluster (i.e., used conditions) and their background (i.e. available conditions). To remove the effect of pseudo-replication in the tracking data, the D statistic and $p-value$ of the KS test are calculated with a degree of freedom based on the number of trajectories in the cluster and not the number of data points (Bohrer et al., 2012). The *p*-values of the KS tests of fall and spring migration trajectory clusters are shown in Table 5 and Table 6, respectively. At 5% significance level, a $p-value$ less than 0.05 indicates that the environmental variable distribution of that cluster is significantly different from that of the background.

Table 5. $p-value$s of Kolmogorov-Smirnov (KS) tests of fall migration trajectory clusters against the background.[2]

Cluster	Thermal uplift	Orographic uplift	NDVI	Temperature	Precipitation	Tailwind	Crosswind
1	$\mathbf{<.001}*$	0.332	$\mathbf{0.002}*$	$\mathbf{<.001}*$	$\mathbf{<.001}*$	$\mathbf{0.002}*$	$\mathbf{<.001}*$
2	0.058	0.611	$\mathbf{<.001}*$	$\mathbf{<.001}*$	$\mathbf{<.001}*$	$\mathbf{0.044}*$	$\mathbf{0.010}*$
3	0.679	$\mathbf{0.040}*$	$\mathbf{<.001}*$	$\mathbf{<.001}*$	0.442	0.625	0.532

[2] Trajectories in cluster 4 and 8 are missing environmental variables, and thus those clusters are not presented in the table.

5	$\mathbf{0.030}$*	0.487	$<\mathbf{.001}$*	$<\mathbf{.001}$*	$<\mathbf{.001}$*	0.082	0.057
6	$\mathbf{0.035}$*	0.348	$<\mathbf{.001}$*	$<\mathbf{.001}$*	$\mathbf{0.002}$*	0.215	0.065
7	$<\mathbf{.001}$*	0.449	$<\mathbf{.001}$*	$<\mathbf{.001}$*	$\mathbf{0.029}$*	$\mathbf{0.002}$*	0.057
9	$<\mathbf{.001}$*	0.802	$<\mathbf{.001}$*	$<\mathbf{.001}$*	$<\mathbf{.001}$*	0.102	$\mathbf{0.007}$*

* denotes that the environmental variable distribution of that cluster is significantly different from that of the background at 5% significance level.

Table 6. $p-value$s of Kolmogorov-Smirnov (KS) tests of spring migration trajectory clusters against the background.[3]

Cluster	Thermal uplift	Orographic uplift	NDVI	Temperature	Precipitation	Tailwind	Crosswind
2	0.370	0.123	$<\mathbf{.001}$*	$<\mathbf{.001}$*	0.336	$\mathbf{0.005}$*	0.220
3	$\mathbf{0.003}$*	0.116	$<\mathbf{.001}$*	$<\mathbf{.001}$*	$<\mathbf{.001}$*	$<\mathbf{.001}$*	$\mathbf{0.002}$*
4	$<\mathbf{.001}$*	0.235	$<\mathbf{.001}$*	$<\mathbf{.001}$*	$<\mathbf{.001}$*	$\mathbf{0.006}$*	$\mathbf{0.004}$*
5	$\mathbf{0.003}$*	0.930	$<\mathbf{.001}$*	$<\mathbf{.001}$*	0.059	$<\mathbf{.001}$*	0.097
6	0.092	$\mathbf{0.027}$*	$<\mathbf{.001}$*	0.072	0.335	$\mathbf{0.002}$*	$\mathbf{0.003}$*
7	$\mathbf{0.001}$*	$\mathbf{0.010}$*	$<\mathbf{.001}$*	$<\mathbf{.001}$*	$<\mathbf{.001}$*	$\mathbf{0.048}$*	0.486
8	$<\mathbf{.001}$*	0.881	$<\mathbf{.001}$*	$<\mathbf{.001}$*	$<\mathbf{.001}$*	0.168	0.129
9	$\mathbf{0.049}$*	0.297	$<\mathbf{.001}$*	$<\mathbf{.001}$*	$\mathbf{0.016}$*	$\mathbf{0.016}$*	$\mathbf{0.043}$*
10	$\mathbf{0.028}$*	$\mathbf{0.011}$*	$<\mathbf{.001}$*	$<\mathbf{.001}$*	$<\mathbf{.001}$*	$<\mathbf{.001}$*	$<\mathbf{.001}$*
11	$<\mathbf{.001}$*	$\mathbf{0.002}$*	$<\mathbf{.001}$*	/	$<\mathbf{.001}$*	$\mathbf{0.005}$*	0.732
12	$\mathbf{0.038}$*	0.272	$<\mathbf{.001}$*	$<\mathbf{.001}$*	$<\mathbf{.001}$*	$\mathbf{0.004}$*	$<\mathbf{.001}$*

* denotes that the environmental variable distribution of that cluster is significantly different from that of the background at 5% significance level. "/" represents the lack of enough data to yield a result.

To further evaluate how the environmental variables influence turkey vultures' migration path, the environmental variables are correlated with the movement speed of the birds corresponding to the tracking points in each cluster using a linear regression. The results suggest that movement speed is significantly associated with thermal uplift, temperature, tailwind, and

[3] Trajectories in cluster 1 are missing environmental variables, and thus those clusters are not presented in the table.

crosswind during the fall migration, and with thermal uplift, NDVI, temperature, and tailwind during the spring migration. This suggest that both wind and thermal uplift are important drivers assisting turkey vulture flights. Further investigation is needed to identify the influence of NDVI.

The distributions of environmental variables of each trajectory cluster in fall and spring are demonstrated in Figure 11 and Figure 12, respectively. Striking between-cluster differences can be observed. This demonstrates that the preferences for environmental variables, especially temperature and NDVI, are disparate among different clusters representing different birds in different seasons. This is confirmed by pairwise KS tests, in which for temperature and NDVI, almost all clusters are distinct at the 5% significance level.

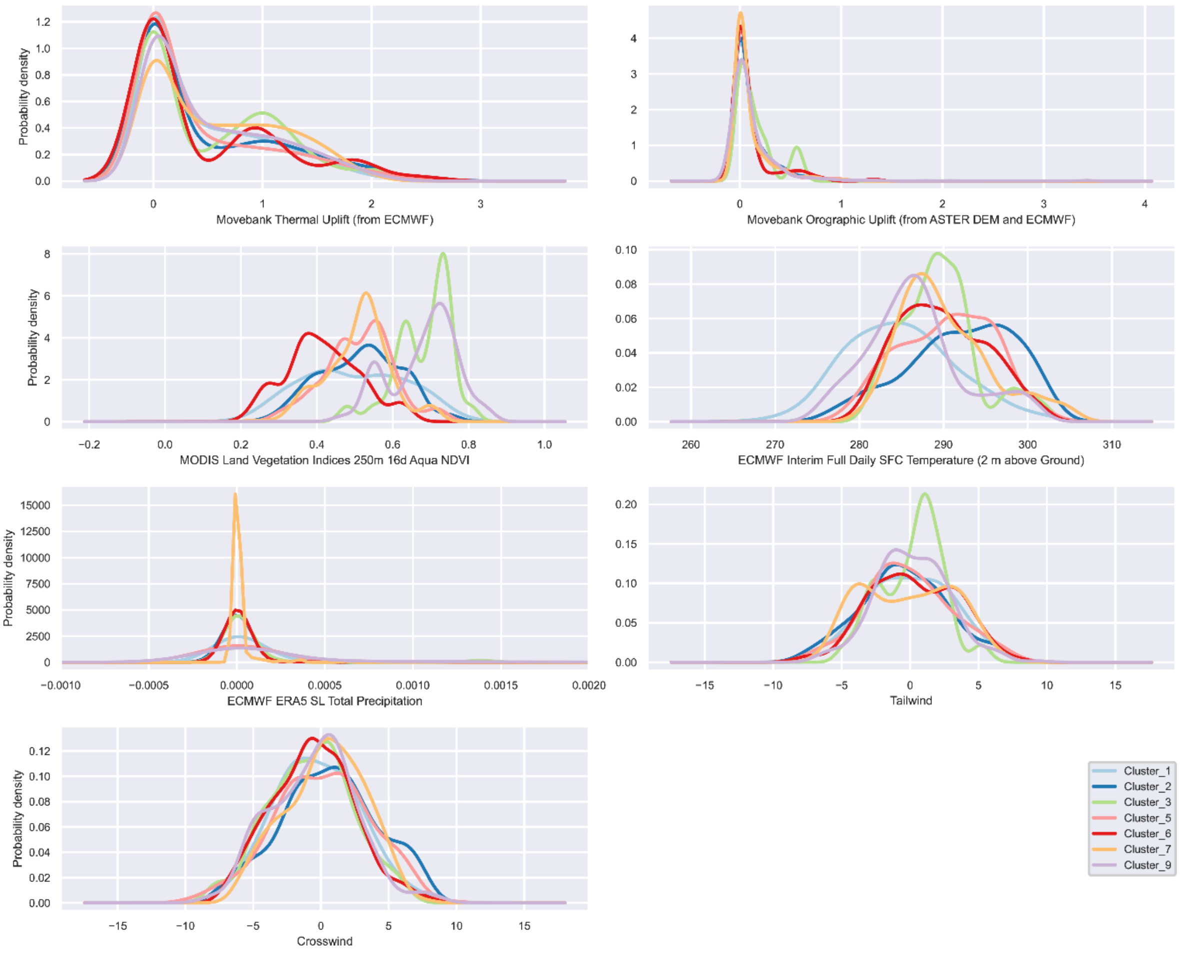
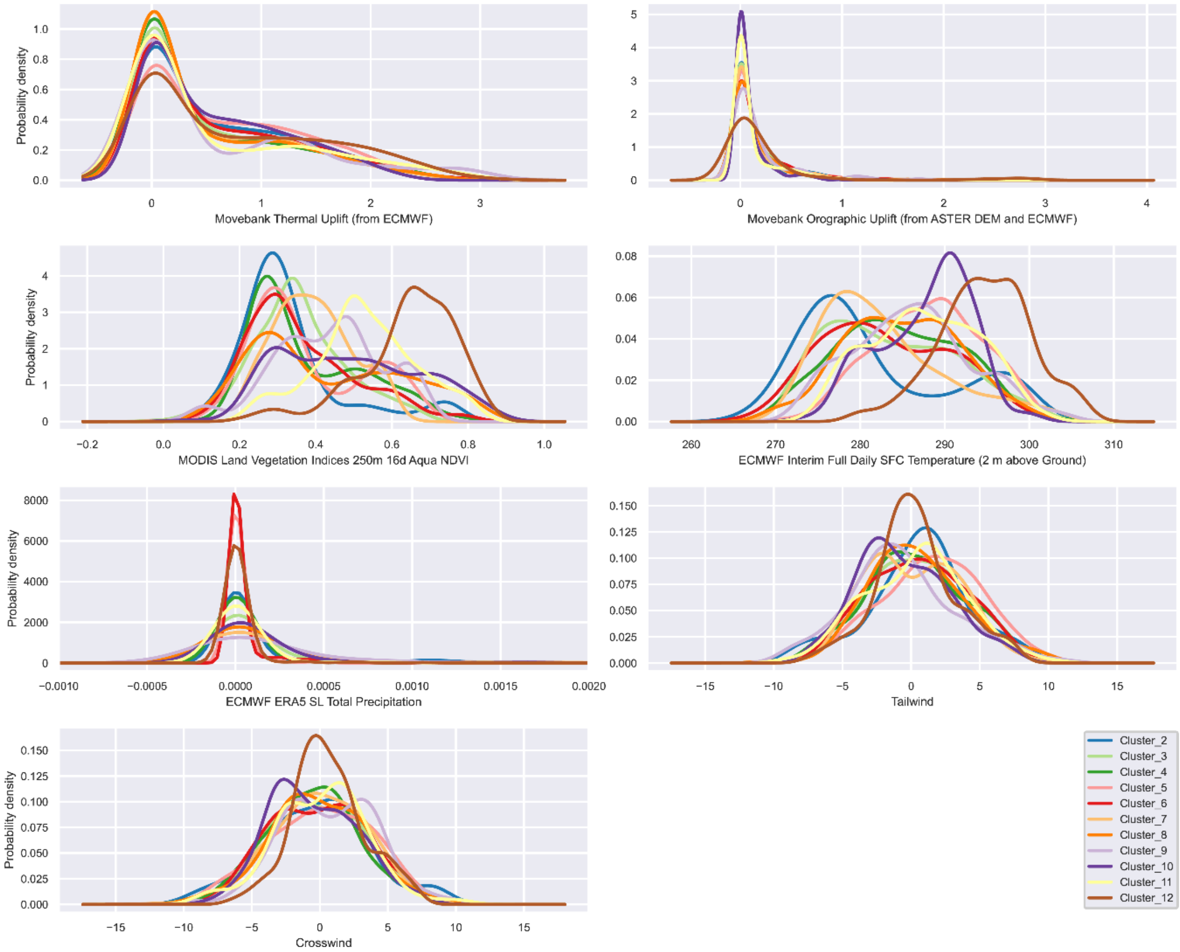


Figure 11. Fall cluster characterization with environmental variables[4]



(b) Spring migration

Figure 12. Spring cluster characterization with environmental variables[5].

Moreover, from the hierarchical clustering results (see Table 3 and Table 4), one can tell that the migration trajectories of the same turkey vulture over different years could fall into different clusters. For example, *Steamhouse 2*'s spring migration paths vary over different years: cluster 11 in 2010, cluster 9 in 2011, cluster 4 in 2012, and cluster 8 in 2013. *Steamhouse 1*'s spring migration paths fall into cluster 3 in 2010 and cluster 7 in 2011. For *Leo*, the turkey vulture

[4] Trajectories in cluster 4 and 8 in fall migration are missing environmental variables in the Env-DATA system on Movebank, thus not plotted.

[5] Trajectories in cluster 1 in spring migration are missing environmental variables in the Env-DATA system on Movebank, thus not plotted.

with the most observations, the spring migration paths are associated with cluster 1 in 2017 and 2020, cluster 3 in 2019, cluster 4 in 2008 and 2010, cluster 5 in 2015 and 2018, cluster 6 in 2013 and 2016, cluster 7 in 2011, cluster 8 in 2012, and cluster 9 in 2019. The migration paths for the same turkey vulture over different years could fall into different clusters, which demonstrates different NDVI and temperature experienced along the paths. This outcome indicates that even for the same turkey vulture, its preference for certain environmental variables is not strictly consistent in different years, which leads to spatial variability in its migration path.

To further investigate this, the Jensen-Shannon distance (JSD) is utilized to analyze the distributional difference of NDVI and temperature with respect to those three turkey vultures. JSD is a metric used to measure the similarity between two probability distributions (e.g., two clusters in this case) (Seideman et al., 2014). The smaller the JSD, the more similar two distributions are. This metric is used to quantify the between-cluster differences, and to investigate whether the same bird would preserve a similar preference of environmental variables compared to those in other clusters. To do so, the pairwise JSD between clusters containing migration paths of the same bird is compared to the average between-cluster JSD (see detailed JSD values in Appendix) using Equation (12-13). This analysis is performed on NDVI and temperature corresponding to the spring migration trajectories of *Steamhouse 1*, *Steamhouse 2*, and *Leo*. Because first, those three turkey vultures have the most tracking data in the dataset. Second, spring migration trajectories show more diverse clustering results than fall. Third, NDVI and temperature demonstrate the most between-cluster differences. The results are shown in Table 7.

$$\overline{JSD} = \frac{\sum_{i=1}^{k}\sum_{j=1,j\neq i}^{k} JSD\left(c_i, c_j\right)}{k(k-1)} \tag{12}$$

$$Percentage = \frac{\sum_{i=1}^{q}\sum_{j=1,j\neq i}^{q} [\![JSD\left(c_{bi}, c_{bj}\right) < \overline{JSD}]\!]}{q(q-1)} \times 100\% \tag{13}$$

where k denotes the number of clusters in spring migration. Let $\{c_{b1}, c_{b2}, \ldots, c_{bq}\}$ denote clusters containing trajectories of bird b. q denotes the number of clusters corresponding to bird b. $[\![*]\!]$ is an Iverson bracket. $[\![P]\!] = \begin{cases} 1, \ if\ P\ is\ true \\ 0, \ if\ P\ is\ false \end{cases}$.

Table 7. Results of individual preference analysis based on Jensen-Shannon distance (JSD)

Turkey vulture	NDVI	Temperature
Steamhouse 1	100%	100%
Steamhouse 2	50%	83%
Leo	86%	81%

4.4 Discussion

This section discusses and interprets the results presented in the previous section. From Table 5 and Table 6, we can tell that the orographic uplift distributions of almost all fall and spring migration trajectory clusters are remarkably similar to the distribution of the background. This indicates that turkey vultures do not follow a specific pattern of orographic uplift during migration. However, for most clusters, almost all other environmental variables have a different distribution compared to that of the background. This suggests that the turkey vultures' choices along their migration trajectories may be intended instead of random, and all those environmental variables except orographic uplift might have contributed to their migration trajectory choosing. In both fall and spring migration seasons, about half of the trajectory clusters suggest that their distributions of crosswind are significantly different from random samples of the background. For tailwind, however, the test results are inconsistent in fall and spring. About half of the fall trajectory clusters yield a significant result, while, interestingly, almost all spring trajectory clusters (except cluster 8) do. This might suggest that slight variation exists in preferences of the environmental conditions in different migration seasons. More significant test results related to tailwind in spring are consistent with the finding that turkey vultures migrate faster on return versus outbound migration (Dodge et al., 2014). The potential reasons behind it include that there is less feeding time as fewer food resources are available in spring (Dodge et al., 2014), and that return (spring) migration includes only experienced birds but no first-time migrants (Bildstein, 2006). Meanwhile, it might also indicate that crosswind is less important than tailwind for turkey vultures. The KS test results, in general, indicates that turkey vultures might vary their migration paths over different years intentionally based on their surrounding environment and in turn, proves the effectiveness of the proposed trajectory clustering framework.

The results of JSD analysis suggest that 50% of the JSDs between each pair of clusters containing *Steamhouse 2* are smaller than the average JSD in terms of NDVI and 83% in terms of temperature. All JSDs between each pair of clusters containing the migration paths of *Steamhouse 1* are smaller than the average JSD in terms of NDVI and temperature, respectively. 86% of the JSDs between each pair of clusters containing *Leo*'s migration paths are smaller than the average JSD in terms of NDVI and 81% in terms of temperature. In general, these percentages are high. These results indicate that even though turkey vultures manifest diverse preferences for some environmental variables and even though the preference of the same turkey vulture is not strictly consistent over different years, each individual tends to preserve a more similar preference over different years, compared to the preferences of other turkey vultures.

One major limitation of the study is that even though 15 years of tracking data are used, the total number of trajectories is not very large, and that many turkey vultures only appear in one or two years in the dataset. This study could be extended by incorporating more sufficient tracking data collected over a long period.

Chapter 5

Trajectory interpolation: filling in the gaps in movement data using machine learning

On the foundation of implicit movement similarity, Chapter 5 introduces a GRU-based trajectory interpolation model with the encoder-decoder architecture to learn both joint and individual movement patterns. A gap contained in a trajectory is to be filled in based on both these joint and individual movement patterns.

5.1 Trajectory interpolation in a raster space

In many cases, trajectories may contain gaps, i.e., missing tracking points in the middle, which need to be interpolated before further movement analysis. Figure 13 presents a schematic illustration of trajectory interpolation. In a typical interpolation scenario, valid tracking points are observed at both ends of a trajectory, while in between, there exists a gap with missing data. The goal of trajectory interpolation is to interpolate those missing tracking points as close to the reality as possible. Note that the ground truth of missing tracking points and trajectory segment to interpolate is shown in Figure 13 for explanatory purposes, while in reality, it is unknown as the corresponding tracking data are missing.

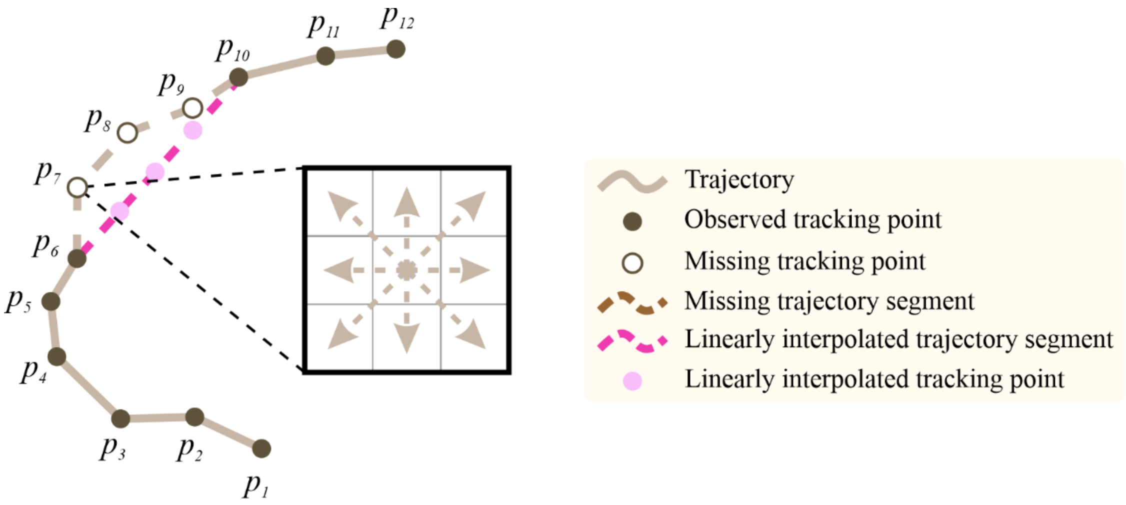


Figure 13. A schematic illustration of trajectory interpolation. The magenta line and points represent linear interpolation as an example interpolation method.

While there are vector-based approaches to trajectory interpolation, this study focuses on trajectory interpolation in a raster space with a constant grid cell width w. The raster space is widely adopted in movement modeling (Ahearn et al., 2017; Grant et al., 2018) and in trajectory interpolation and prediction particularly (Hirakawa et al., 2018; Qin et al., 2021; Rew et al., 2019). The primary advantage is that using rasterized trajectories significantly reduces the computational complexity. At each time step, the moving entity moves to one of the eight neighborhood grid cells of its previous location (see Figure 13). In this way, the location interpolation is converted to a classification problem, while the interpolation of other attributes, such as time, speed, and heading, remains a regression problem. Another advantage of using the raster structure is that the model can be further enhanced and informed by environmental variables deriving movement as environmental data often are available in a raster format.

5.2 Method

In trajectory interpolation, to make well informed predictions, the most important knowledge comes from implicit movement similarity in the observed tracking points of that very trajectory as well as of other trajectories for the same or similar moving entity. On that foundation, this study proposes a trajectory interpolation model with an encoder-decoder architecture based on gated recurrent units (GRUs) (Cho et al., 2014). Figure 14 demonstrates the architecture of the proposed sequence-to-sequence trajectory interpolation model, including a spatiotemporal feature extraction module, an encoder, a decoder, fully connected layers, and finally, a tracking data retrieval module.

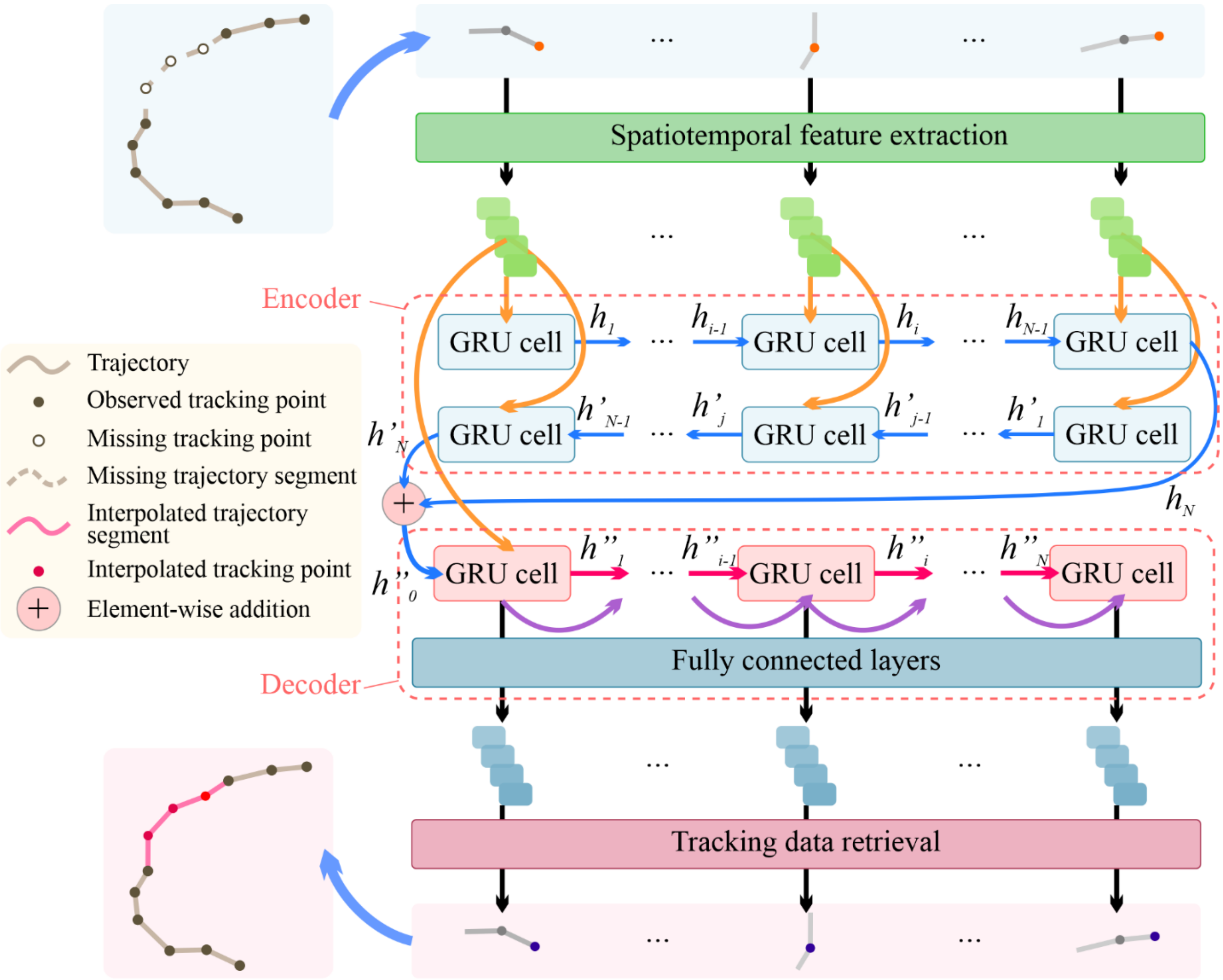
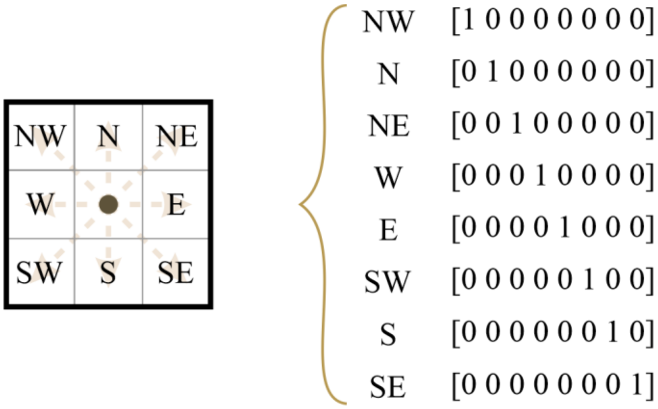


Figure 14. Architecture of the proposed trajectory interpolation model based on GRU

The spatiotemporal feature extraction module converts a rasterized trajectory composed of tracking points $T = \{p_1, p_2, \dots, p_N\}$ into a sequence of spatiotemporal feature vectors $\{f_1, f_2, \dots, f_N\}$ that can be processed by the encoder. Feature vector f_i $(1 \leq i \leq N)$ stores information concerning the cell location, time step, speed, and heading corresponding to tracking point p_i. In this study, movement is represented as the relative locational change with respect to the previous point in the raster, which is an approach commonly used in movement modeling (Hirakawa et al., 2018; W. Li et al., 2019; Qin et al., 2021). Using relative locational changes helps the model focus on the movement patterns instead of the absolute coordinate values themselves, thus giving the model better prediction capacity. The first component of f_i $(1 < i \leq N)$ represents the relative locational change of p_i with respect to p_{i-1}. The eight types of possible locational changes $\{NW, N, NE, W, E, SW, S, SE\}$ are mapped into vectors using one-hot encoding (see Figure 15). Since the first point in a trajectory has no previous points, its corresponding locational change vector has zeros only. The second component of f_i $(1 \leq i \leq N)$ is time step, which is calculated based on the time elapsed since the first point of that trajectory. Note that information regarding the relative time internal between consecutive tracking points is preserved in this way, because an RNN model reads data sequentially. The third and fourth components f_i $(1 \leq i \leq N)$ are speed and heading, respectively. Note that all components except the first relative locational change are normalized before input into the encoder.



NW	[1 0 0 0 0 0 0 0]
N	[0 1 0 0 0 0 0 0]
NE	[0 0 1 0 0 0 0 0]
W	[0 0 0 1 0 0 0 0]
E	[0 0 0 0 1 0 0 0]
SW	[0 0 0 0 0 1 0 0]
S	[0 0 0 0 0 0 1 0]
SE	[0 0 0 0 0 0 0 1]

Figure 15. One-hot encoding of relative locational change

The core module of both the encoder and the decoder is GRU. Specifically, the encoder consists of stacked bidirectional GRU layers, while the decoder is a single unidirectional GRU layer with fully connected layers attached. Figure 16 illustrates the structure of GRU, where x_t

and h_t are the input and output of GRU at time t, respectively, and r_t, z_t denote the vectors of the reset gate and update gate. GRU works based on the following mechanism (Cho et al., 2014). When an input $x_t$is passed to a GRU, it first goes through the reset gate together with the previous hidden state h_{t-1} to calculate vector r_t as in Equation (14). The reset gate helps determine which past information is to forget and reset with the current input. Similarly, the input x_t goes through the update gate together with the previous hidden state h_{t-1} to calculate vector z_t following Equation (15). The update gate helps determine the amount of information from the previous hidden state that needs to pass along to the current one. Finally, the current hidden state h_t is calculated based on all these vectors as in Equation (16-17). The reset gate captures short-term dependencies, while the update gate captures long-term dependencies. In this way, GRU possesses the same ability to capture both long-term and short-term dependencies as LSTM, but GRU is able to do so with fewer gates and fewer parameters.

$$r_t = \sigma(W_r) \cdot [h_{t-1}, x_t] \tag{14}$$

$$z_t = \sigma(W_z) \cdot [h_{t-1}, x_t] \tag{15}$$

$$\tilde{h}_t = \tanh(W \cdot [r_t \otimes h_{t-1}, x_t]) \tag{16}$$

$$h_t = (1 - z_t) \otimes h_{t-1} + z_t \otimes \tilde{h}_t \tag{17}$$

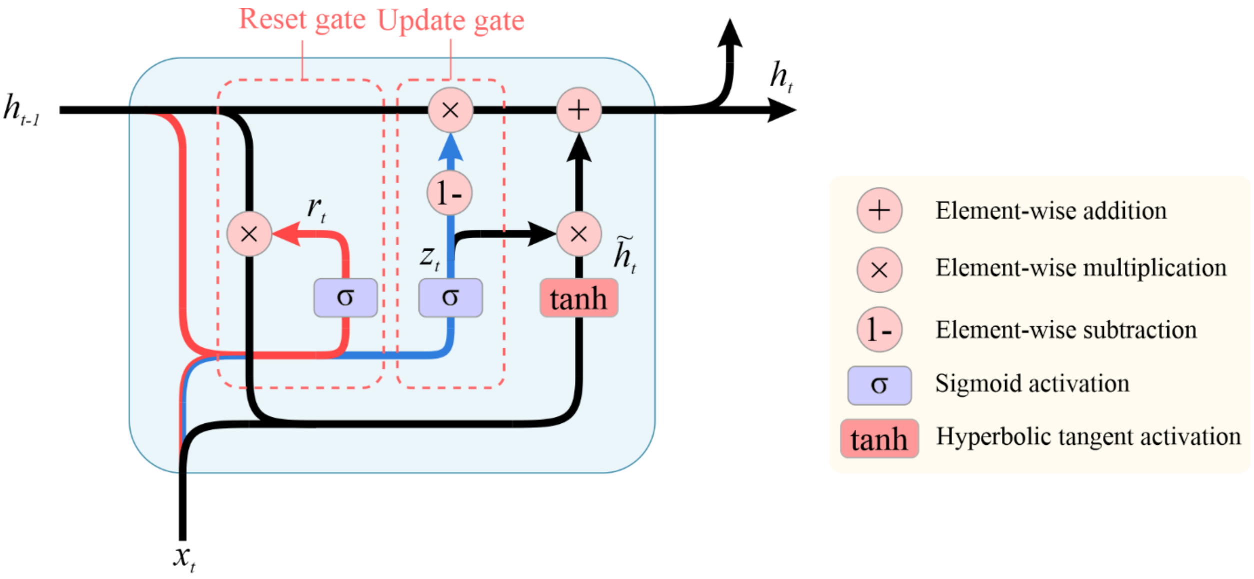


Figure 16. A schematic illustration of a gated recurrent unit (GRU), as proposed by Cho et al. (2014)

The encoder sequentially processes the feature vector sequence $\{f_1, f_2, \dots, f_N\}$, some of which represent missing values, to learn the implicit movement similarity, which will later be used by the decoder to reconstruct the complete feature vector sequence without missing values. Note that the encoder consists of stacked bidirectional GRU layers. The first GRU layer reads $\{f_1, f_2, \dots, f_N\}$ chronologically, while the second GRU layer reads it reversely. In the encoder, all inputs of GRUs come from the feature vector sequence directly, and all GRU output hidden states $\{h_1, h_2, \dots, h_N\}$ and $\{h'_1, h'_2, \dots, h'_N\}$ except the last ones are discarded. The last hidden states of those two stacked layers h_N, h'_N are fused to be the initial hidden state of the decoder h''_0, based on which, the decoder decodes the complete feature vector sequence $\{f'_1, f'_2, \dots, f'_N\}$ without missing values. In specifics, the output of a decoder GRU cell h''_i $(1 \leq i \leq N)$ is passed to two modules of fully connected layers (see Figure 17). Based on h''_i, the first module predicts the relative locational change in the one-hot encoding form and second module predicts other spatiotemporal attributes including time step, speed, and heading. The predictions from these two modules are concatenated to be the final predicted feature vector f'_i. Note that the input and output feature vectors f_i and f'_i have the same form. The only difference is that some feature vectors in the input sequence represent missing values, while no values in the output sequence are missing. In this way, the gap in the trajectory is interpolated.

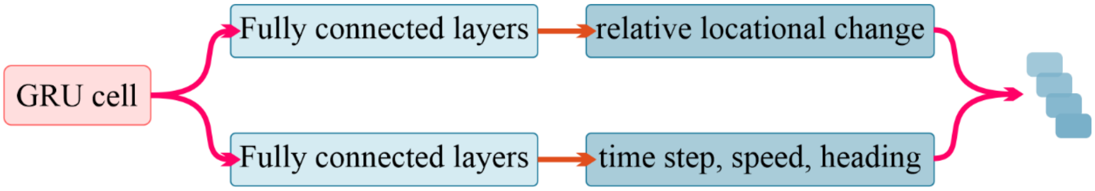


Figure 17. Illustration of the decoder GRU output

The proposed model needs to be trained on trajectories with manually created gaps, and thus the ground truth of the missing values is known during the training process. Once the model is sufficiently trained, it can be applied to trajectories with real gaps. During the training process, both cross-entropy loss and mean-squared-error loss are used. Because the proposed trajectory interpolation model is doing both the classification to predict the relative locational change and regression to predict spatiotemporal attributes including the time step, speed, and heading.

5.3 Experiment

5.3.1 Trajectory dataset and preprocessing

For tracking point p_i, its speed is calculated according to the spatial length of vector $\overrightarrow{p_\imath p_{\imath+1}}$ and the corresponding time interval. The unit of speed in this study is km/h. The heading of p_i is defined as the clockwise angle between due north and vector $\overrightarrow{p_\imath p_{\imath+1}}$. Note that the speed and heading are calculated prior to rasterization to preserve accuracy. The average speed of the migration tracking points is 8.2 km/h with a standard deviation of 15.0 km/h. Therefore, the trajectories are transformed into the raster space with a constant grid cell width $w = 10\ km$. A minimal-area rectangle is created according to the minimum bounding box of the tracking points, representing the boundary of the area that turkey vultures travel in. In order to allow for a certain level of uncertainty, the boundary of the raster cells is defined based on the minimum bounding box with a 5 km buffer. When trajectories are transformed into the raster space, each tracking point is mapped to a raster cell according to spatial proximity. If multiple successive tracking points of the same trajectory are assigned with the same raster cell coordinates, only the last tracking point in that cell is preserved. This guarantees that each trajectory consists of distinct cell positions. Meanwhile, the stopover behavior is still preserved in the trajectory, as one can tell from the time elapsed. In contrast, if two consecutive tracking points in a trajectory are not in adjacent cells, tracking points are interpolated based on a least-cost random walk model (Squires et al., 2013). This step is necessary to ensure that after preprocessing, each tracking point (except the first one) is in the one of the eight neighborhood cell of its previous point in the trajectory (Hirakawa et al., 2018). Trajectories with fewer than 200 tracking points are eliminated since the input sequence length of the trajectory interpolation model is empirically set as $seqLen = 200$. It is worth noting that the proposed trajectory interpolation model works with a sliding window mechanism. It means that the model can still work on trajectories longer than $seqLen$. If it is a long trajectory with only 1 gap, a typical approach is to only input the trajectory segment, whose length is $seqLen$, containing the gap. Analogously, the proposed model can even work on a long trajectory with multiple gaps as long as these gaps are not adjacent to each other. The time step of each tracking point is calculated based on the time elapsed since the first point of that trajectory. After preprocessing, there are 61 trajectories with 43683 tracking points (see Figure 18 for visualization).

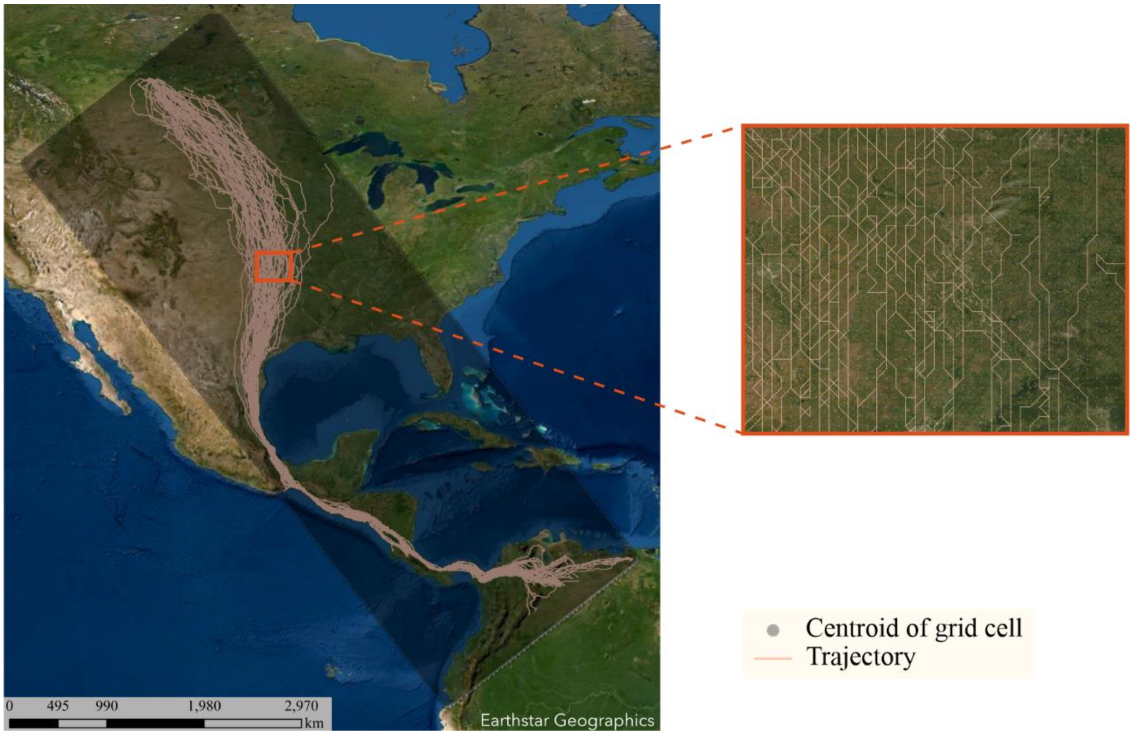


Figure 18. Rasterized trajectories of turkey vulture migration.

5.3.2 Training and evaluation of the proposed model

To train and evaluate the proposed trajectory interpolation model, this study uses trajectory samples containing gaps that are manually created. To simulate the stochasticity of gap lengths, the number of missing tracking points contained in a gap is sampled from a normal distribution with $\mu = 55$ and $\sigma = 10$ to generate a reasonable range of lengths of gaps that are possibly contained in trajectory samples with sequence length $seqLen = 200$ (see Figure 19). For example, if 100000 gaps are generated, the minimum length is 12 while the maximum length is 99. Since the topic of this study is trajectory interpolation instead of extrapolation, there need to be some valid observed tracking points at both ends of a trajectory sample. In this study, the number of observed tracking points at each end is set to 50, while a gap with the length drawn from the normal distribution can start anywhere in between. In this way, 80300 trajectory samples are created based on the actual turkey vulture tracking data. Following the machine learning tradition, the trajectory samples are split into training set, validation set, and test set according to a ratio of 0.8:0.1:0.1. The training set, validation set, and test set contain 64240, 8030, and 8030 trajectory samples, respectively.

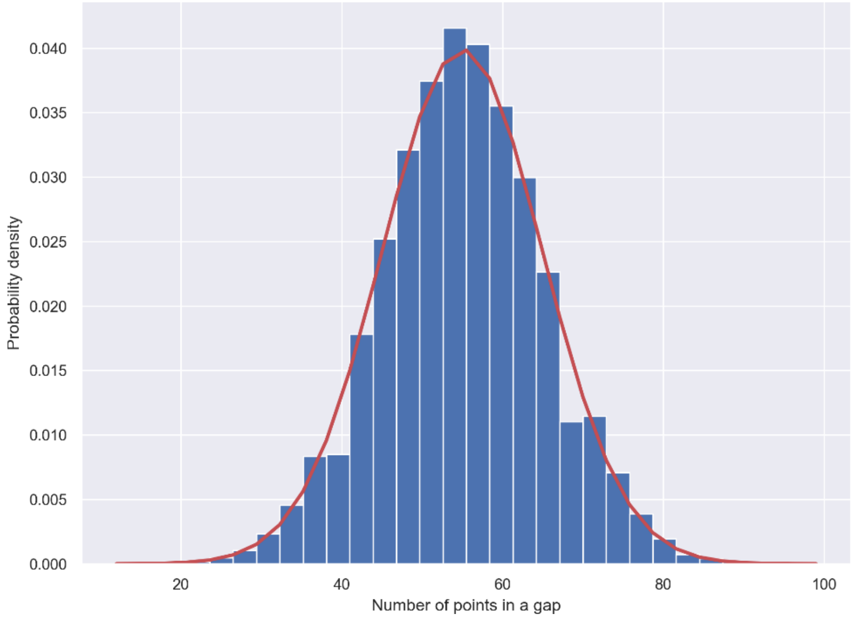
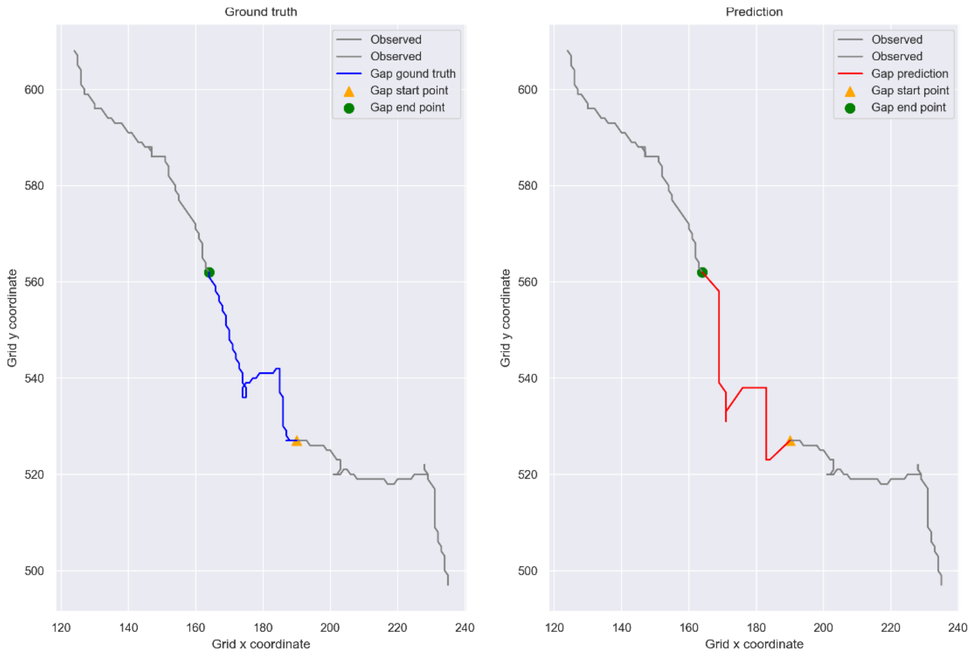


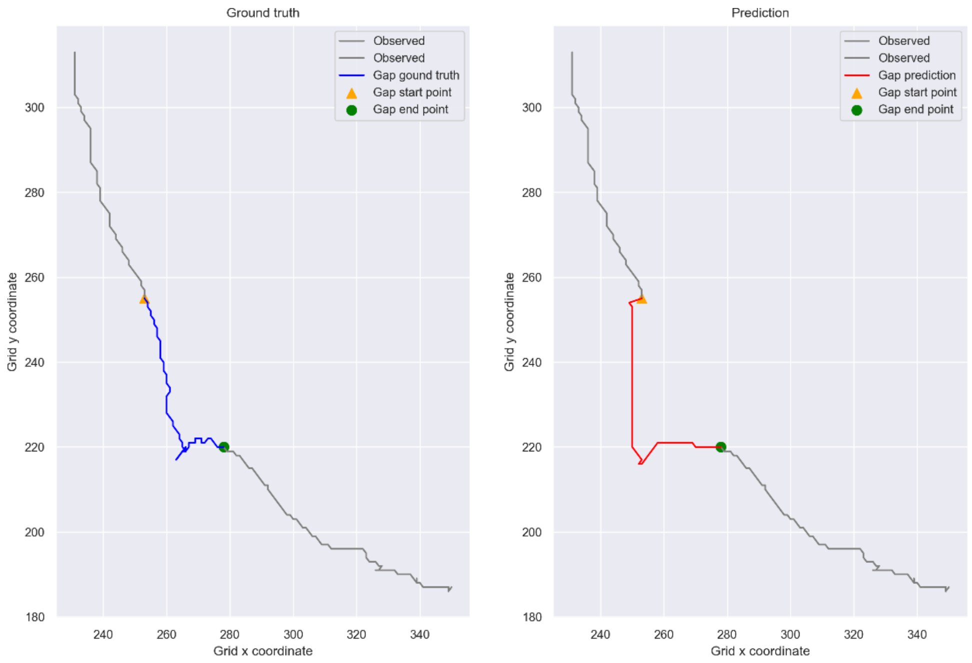
Figure 19. Gap lengths sampled from a normal distribution with $\mu = 55$ and $\sigma = 10$

The proposed trajectory interpolation model is implemented using PyTorch (Paszke et al., 2019), which is an open-source artificial neural network library written in Python. The loss function used to train the model is designed as the summation of the cross-entropy loss, which deals with the location change classification, and the mean-squared-error loss, which deals with the attribute regression. The model is trained with the Adam optimizer with an adaptive learning rate from 1×10^{-3} to 1×10^{-5} using the AMSGrad algorithm (Kingma & Ba, 2015; Reddi et al., 2018). To determine the optimal hyperparameter setting of the proposed model, multiple combinations of hyperparameter settings are experimented on. The hyperparameters of the final model, which has the smallest final validation loss 1.0726 after 100 training epochs, are as follows. Both the encoder and the decoder GRU cells have a hidden size of 128, meaning that the number of features in the hidden state is 128. The encoder consists of stacked bidirectional GRU layers, while the decoder is a single unidirectional GRU layer with fully connected layers with a hidden size of 512 attached. The dropout probability is set to 0.5 to alleviate the overfitting problem (Srivastava et al., 2014). Layer normalization is utilized before the fully connected layers to obtain smoother gradients, and it is helpful for faster training and better generalization capabilities (Xu et al., 2019).

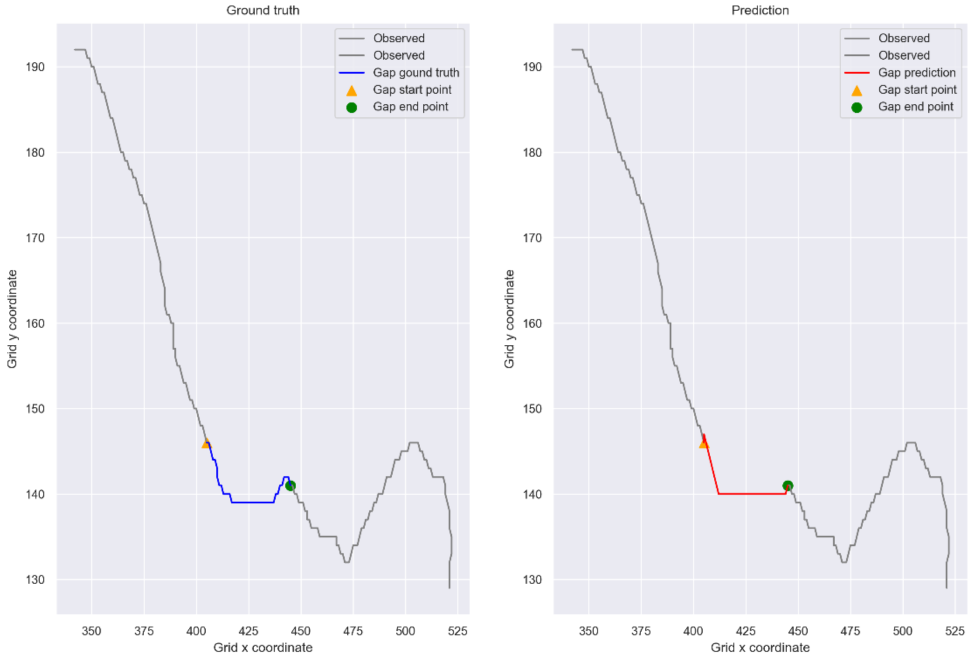
Figure 20 demonstrates the interpolation results of several example trajectories in the test set using the proposed GRU-based model. The results suggest that the proposed GRU-based interpolation model is capable of learning implicit movement similarity contained in a trajectory, since without which, some gaps would be difficult to interpolate accurately using traditional interpolation methods. For example, in Figure 20(a), the trajectory segment in the gap has an unusual geometric shape, thus making the interpolation extremely challenging, but the proposed model successfully learns enough information to reconstruct it. However, these interpolation results also suggest that the proposed GRU-based model is underestimating the sinuosity and stochasticity of trajectories. Take Figure 20(c) as an example. In the gap area, the turkey vulture first moves southeast for a while, then its heads east, and finally it moves northeast a little till the end of the gap. The general geometric shape of the interpolated results follows that patten, but the model uses mostly straight line segments, leading to oversimplification of the geometry with respect to the interpolated trajectory segment. Another problem is that the model tends to exaggerate some patterns. In Figure 20(b), for instance, the object does not head towards the gap end point directly but lingers in an area west to the gap end point. This lingering behavior is successfully predicted by the model, but there is quite a deviation in the prediction of the lingering location. Similarly, in Figure 20(d), the turkey vulture does not head southeast directly but first flies southwest for a short time and then goes back. These suggest that the model is inclined to exaggerate the magnitude of a behavior that drives the moving entity away from the beeline between the start and end points of a gap. The same inclinations can also be found in Figure 20(e-f). In a word, the performance of the model is not very stable. It can generate surprisingly satisfactory results, while it may also fail with large deviations from the actual path. These discussed problems may suggest that the proposed model needs a better training strategy including adding other variables (e.g., path tortuosity and environmental variables) and probably a larger trajectory sample to increase its effectiveness and robustness. Due to the complexity that comes with the architecture of the proposed GRU-based trajectory interpolation model, training this model requires more attention and further study, as with many deep neural network models (Cyr et al., 2020; Glorot & Bengio, 2010; Han et al., 2018).



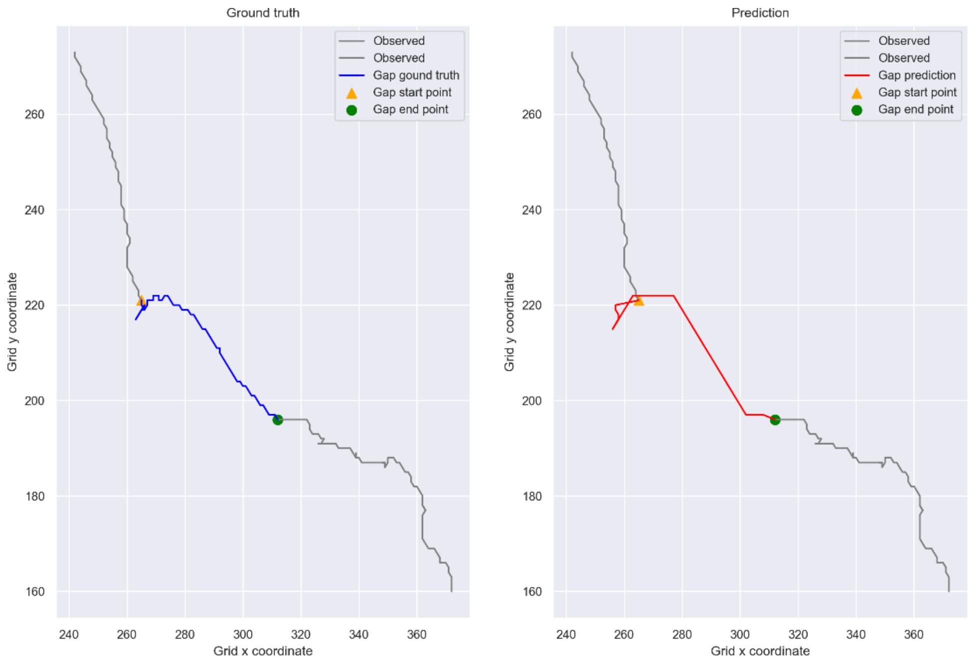
(a) Test trajectory sample with id 1



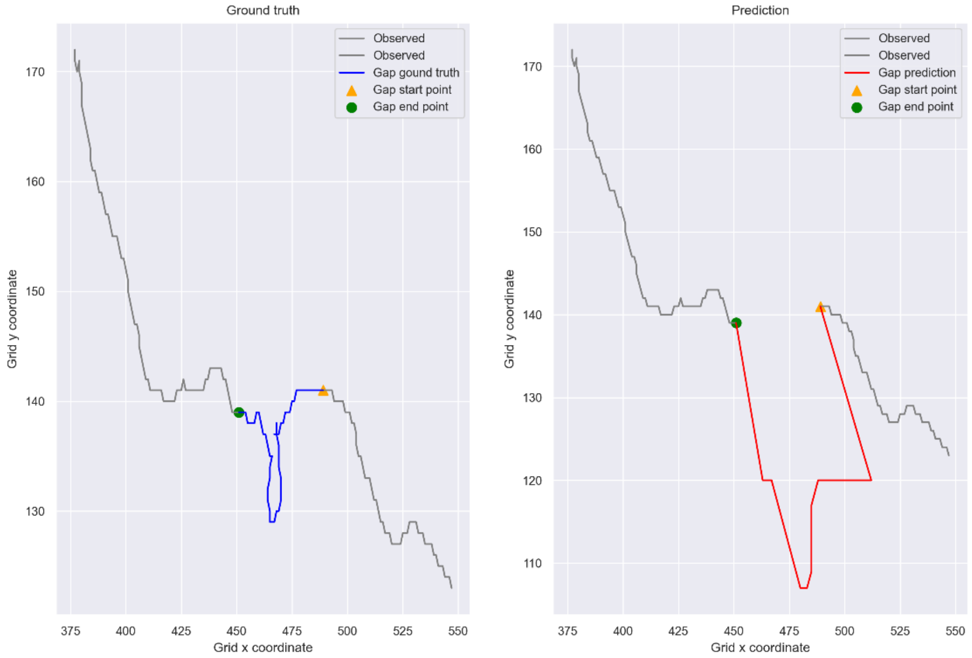
(b) Test trajectory sample with id 212



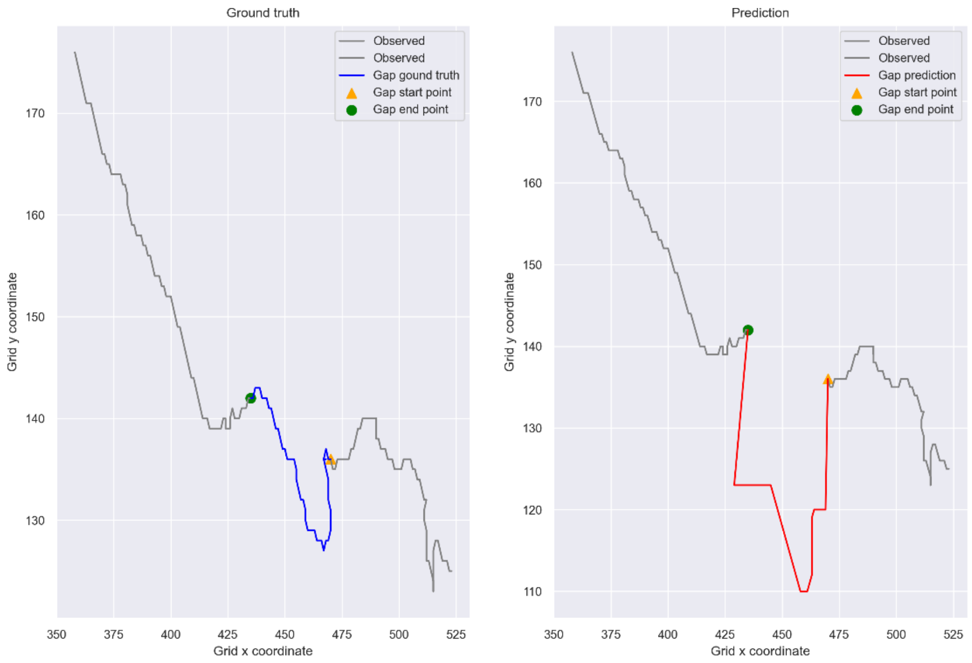
(c) Test trajectory sample with id 4944



(d) Test trajectory sample with id 4236



(e) Test trajectory sample with id 28



(f) Test trajectory sample with id 2110

Figure 20. Interpolation results of several example trajectories in the test set. In each subfigure, the ground truth is shown the left, while the interpolation result using the proposed model is on the right. The observed parts, visualized in grey, exist on both ends of a trajectory, since this is an interpolation task. The yellow triangle denotes where the gap starts, and the green circle denotes where the gap ends.

5.4 Discussion and future work

The proposed trajectory interpolation model is capable of reading a trajectory containing a gap in both chronological and reverse chronological orders. The information obtained from these two directions is fused to learn the implicit movement similarity contained in that trajectory, which is later used to reconstruct a complete trajectory without the gap. The capability of the proposed interpolation model primarily comes from this architecture. One aspect of the future work is to incorporate context information, as it is an important driver that influences the movement path selection in both animal movement and human movement. This includes environmental context (e.g., preferences, physical constraints) and social context (e.g., interactions with other individuals nearby) (Bohrer et al., 2012; Dodge et al., 2014; Gupta et al., 2018; Sadeghian et al., 2019; Shi et al., 2019). This requires a more powerful model that is capable of processing multiple trajectories simultaneously and modeling agent interactions. And the model needs to be parameterized and validated on tracking data from different types of moving entities, such as animals, humans, and vehicles.

One major limitation of the proposed trajectory interpolation method is that transforming tracking data from the vector space to the raster space decreases the data precision, the extent of which depends on the chosen grid cell width w. The larger the w, the higher the precision decrese. However, the computational cost increases dramatically if a smaller w is chosen. Because it means that the study area is divided into considerably more cells with finer resolution. This is a trade-off between computational cost and precision. With the development of modern computational resources, one future direction of this study is to refine the proposed methodology to make it suitable for large-scale parallel and distributed computing.

Gated recurrent unit (GRU) is a newer generation of the recurrent neural network (RNN) compared to long short-term memory (LSTM). GRU gets rid of LSTM’s cell state and use only the hidden state to transfer information. By doing so, it decreases the number of gates and parameters needed, which makes it easier to train and it can have a similar or even better performance compared to LSTM, especially with a sample size that is not very large (Cho et al., 2014). However, it is still worth further study whether and to what extent the tracking dataset size influences the performance of the models based on GRU vs. models based on LSTM. With the rapidly development of tracking technology and the widespread of tracking devices, tracking data do come in enormous volume in many cases.

Moreover, when transforming trajectories from the vector space to the raster space, grid cells are used, since grid cells are the most widely adopted cell shape in rasterization. But more attention could be paid to whether and how changing the shape, to hexagon for example, influences the interpolation effectiveness and efficiency.

Chapter 6

Conclusion

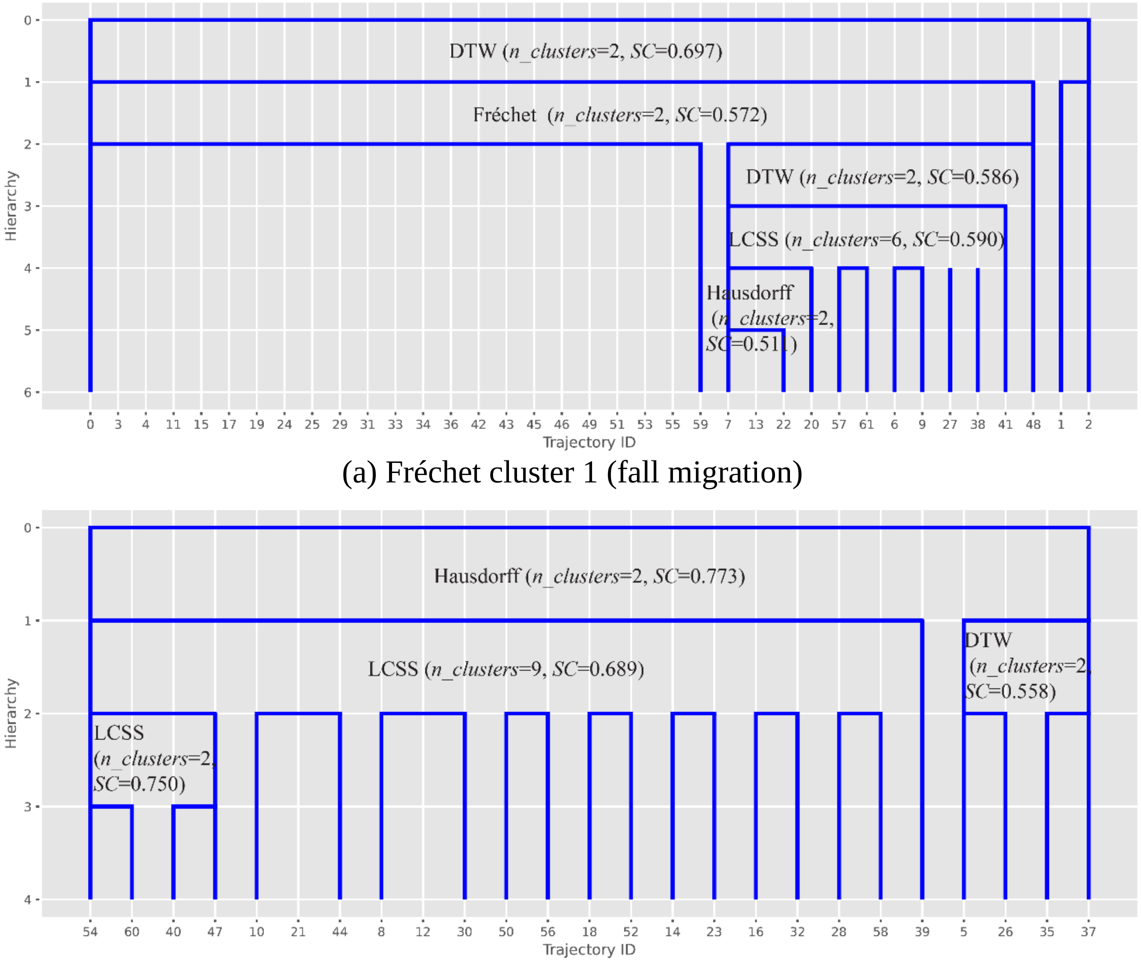
Movement similarity, classified into explicit and implicit movement similarity in this thesis, is a promising field and research direction in computational movement analysis. In terms of explicit movement similarity, this thesis explores an integrated hierarchical clustering framework to analyze movement ecology. Multiple commonly-used trajectory similarity measures including Fréchet distance, DTW, Hausdorff distance, LCSS, and NWED, a special kind of edit distance, are utilized collectively. At each hierarchy, the optimal clustering setting, i.e., a distance metric together with the number of clusters, is chosen automatically based on silhouette coefficient (SC). Then, the environmental drivers that lead to variation are inferred by comparing tracking data samples against random samples collected from the study area. To assess the applicability of the proposed framework, this study investigates similarity patterns in long-term trajectories of migratory turkey vultures *(Cathartes aura)* in North America using tracking data during 15 years of fall and spring migrations seasons and infer the environmental drivers that contribute to turkey vulture's migration path choosing. While orographic uplift is not critical to turkey vultures, all other environmental factors including thermal uplift, temperature, vegetation, wind patterns, and precipitation seem to matter. However, various clusters of migration trajectories manifest that turkey vultures might not be following the same pattern of specific environmental factors. Understanding how animals respond to environmental factors when choosing migration paths is critical to both animal conservation and trajectory prediction with predicted environmental conditions taking into account (Shamoun-Baranes et al., 2010). Moreover, it may in turn provide insights into evaluating and predicting certain environmental conditions based on animals' ecological responses.

As for implicit movement similarity, this thesis proposes a trajectory interpolation model with an encoder-decoder architecture based on gated recurrent units (GRUs) to interpolate trajectory gaps (missing values) in the raster space. The capability of the proposed interpolation model primarily comes from this architecture, enabling it to outperform several existing interpolation methods. A trajectory containing a gap is first converted into a multi-dimensional feature vector sequence before input into the encoder. The encoder consists of stacked bidirectional GRU layers, and thus is capable of reading the trajectory containing a gap chronologically and reversely. And the structure of GRU allows it to capture both long-term and short-term dependencies. The information obtained from these two directions is fused to learn the implicit movement similarity contained in that trajectory, which is later used to reconstruct the complete trajectory without the gap.

In a nutshell, on the foundation of explicit and implicit movement similarity, researchers can build multifarious models for diverse moving entities, such as animals, humans, vessels, and vehicles. However, there still remain many challenges and opportunities for future work to consider in this area. Examples include incorporating context information in either explicit or implicit movement similarity and considering both joint and individual movement patterns.

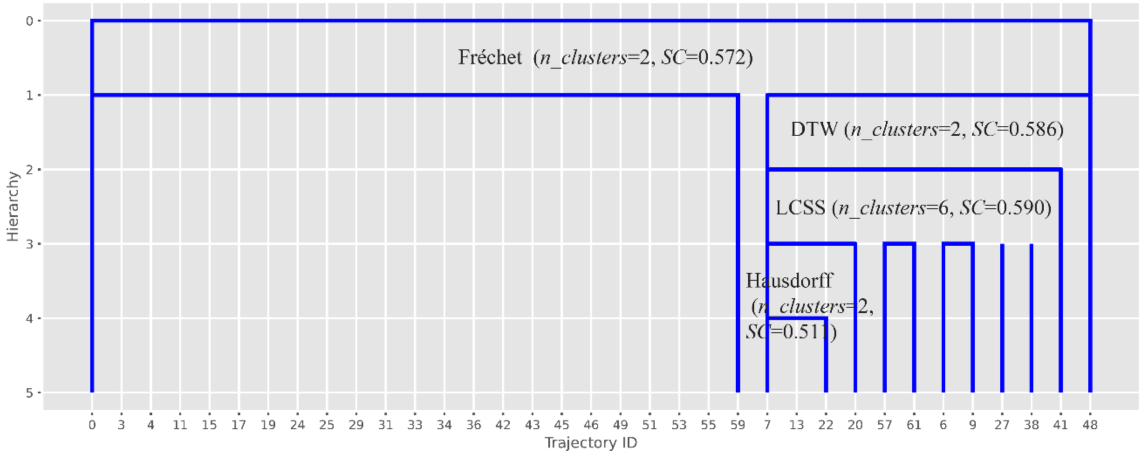
Appendix

The hierarchical clustering framework (proposed in Section 4.1) is applied to initial results of each similarity measure obtained in Section 4.3.1 to test the robustness of the methodology, i.e., whether starting from different initial results would significantly influence the final clustering outcome. Figure 21 illustrates the hierarchical clustering results with various starting measure. Between the adjacent hierarchies, the distance metric used for clustering together with the optimal $n\_clusters$ and the corresponding SC computed is presented.

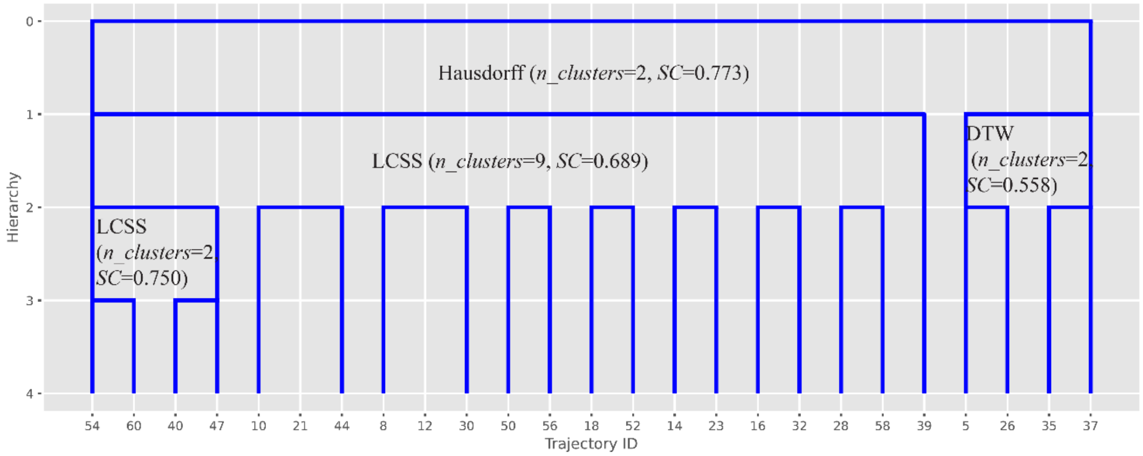


(a) Fréchet cluster 1 (fall migration)

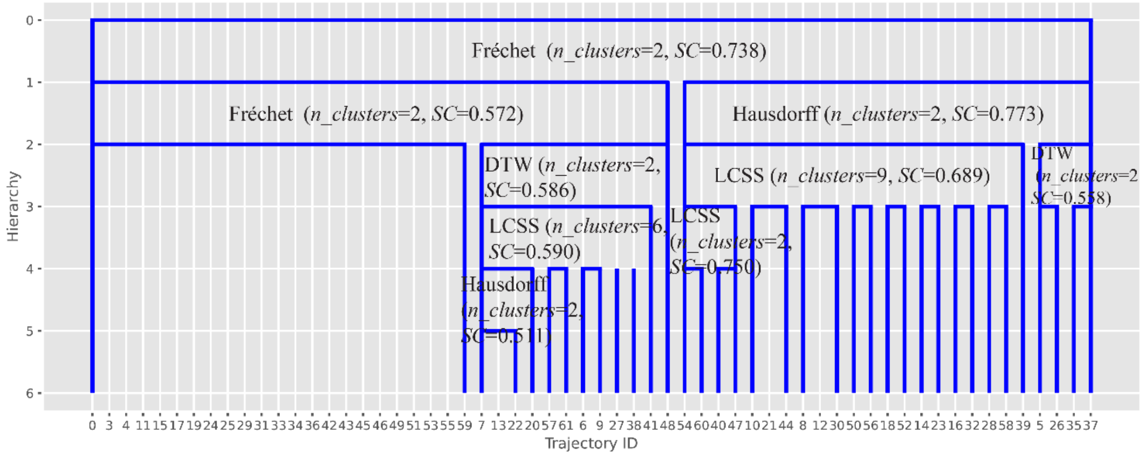
(b) Fréchet cluster 2 (spring migration)



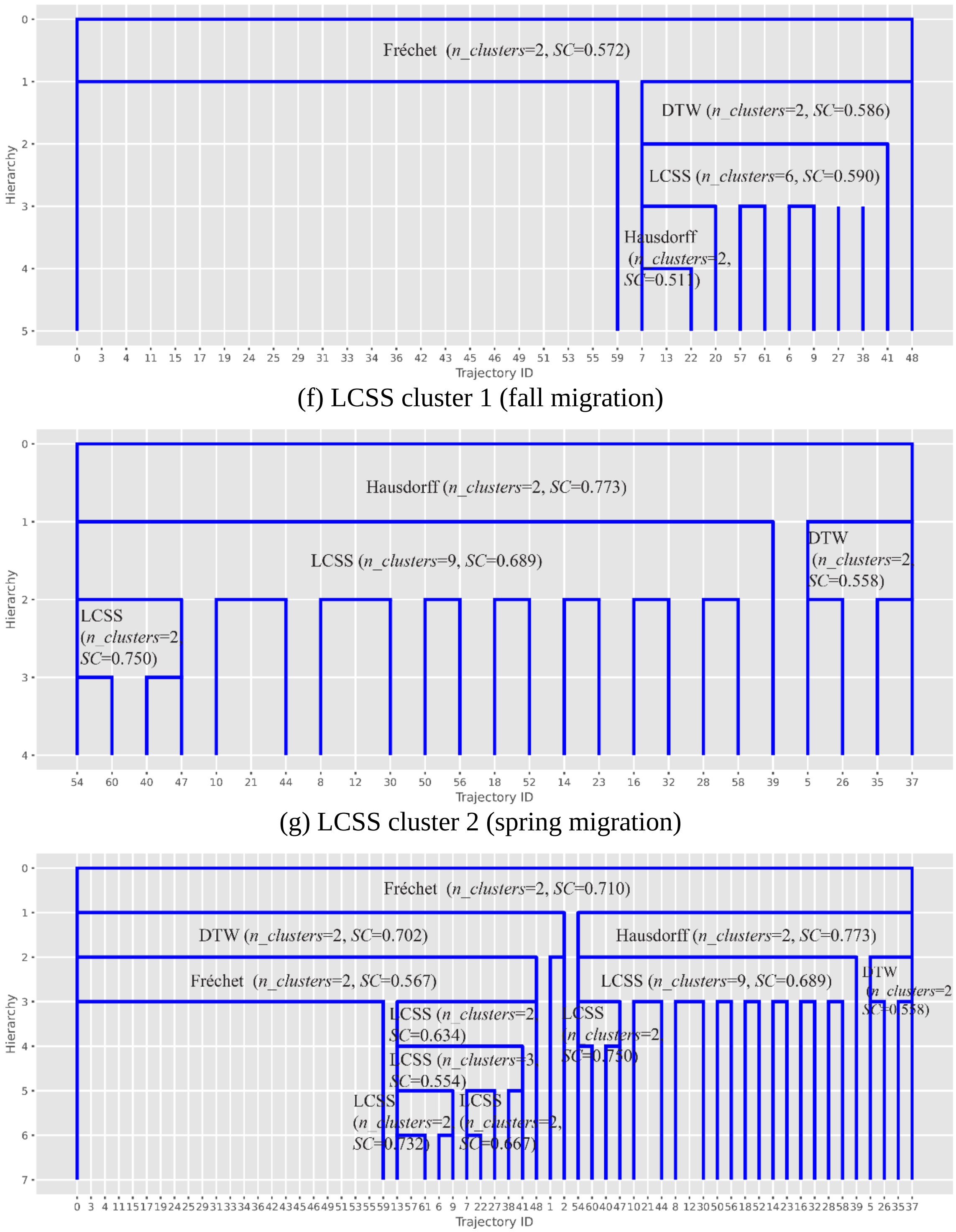
(c) DTW cluster 1 (fall migration)



(d) DTW cluster 2 (spring migration)



(e) Hausdorff cluster 1 (fall and spring migration)



(f) LCSS cluster 1 (fall migration)

(g) LCSS cluster 2 (spring migration)

(h) NWED cluster 1 (fall and spring migration)

Figure 21. Hierarchical clustering results. The title of each subplot indicates the initial cluster obtained previously (see Section 5.1) on which the hierarchical clustering starts. SC denotes the silhouette coefficient and $n\_clusters$ denotes the number of clusters.

The detailed between-cluster Jensen-Shannon distance (JSD) values for spring migration paths in terms of NDVI and temperature are presented in Table 8 and Table 9. JSDs are computed using the SciPy library (Virtanen et al., 2020) with the default logarithm base e.

Table 8. Between-cluster Jensen-Shannon distances (JSDs) for spring migration paths in terms of NDVI[6]

Cluster Cluster	2	3	4	5	6	7	8	9	10	11	12
2	0	0.430	0.423	0.471	0.428	0.507	0.489	0.633	0.560	0.653	0.723
3	0.430	0	0.387	0.442	0.364	0.371	0.442	0.480	0.521	0.560	0.691
4	0.423	0.387	0	0.427	0.345	0.410	0.416	0.514	0.503	0.514	0.654
5	0.471	0.442	0.427	0	0.392	0.473	0.444	0.592	0.518	0.591	0.672
6	0.428	0.364	0.345	0.392	0	0.388	0.402	0.530	0.480	0.542	0.659
7	0.507	0.371	0.410	0.473	0.388	0	0.488	0.511	0.503	0.566	0.701
8	0.489	0.442	0.416	0.444	0.402	0.488	0	0.555	0.472	0.528	0.555
9	0.633	0.480	0.514	0.592	0.530	0.511	0.555	0	0.580	0.558	0.654
10	0.560	0.521	0.503	0.518	0.480	0.503	0.472	0.580	0	0.514	0.549
11	0.653	0.560	0.514	0.591	0.542	0.566	0.528	0.558	0.514	0	0.545
12	0.723	0.691	0.654	0.672	0.659	0.701	0.555	0.654	0.549	0.545	0

Table 9. Between-cluster Jensen-Shannon distances (JSDs) for spring migration paths in terms of temperature[1]

Cluster Cluster	2	3	4	5	6	7	8	9	10	11	12
2	0	0.357	0.390	0.543	0.353	0.345	0.438	0.488	0.578	0.492	0.565
3	0.357	0	0.222	0.389	0.215	0.232	0.273	0.382	0.385	0.296	0.498
4	0.390	0.222	0	0.356	0.207	0.257	0.237	0.370	0.341	0.265	0.509
5	0.543	0.389	0.356	0	0.376	0.410	0.345	0.424	0.340	0.321	0.473
6	0.353	0.215	0.207	0.376	0	0.279	0.258	0.406	0.387	0.343	0.526
7	0.345	0.232	0.257	0.410	0.279	0	0.302	0.393	0.458	0.375	0.561
8	0.438	0.273	0.237	0.345	0.258	0.302	0	0.362	0.335	0.288	0.531

[6] Trajectories in cluster 1 in spring migration are missing environmental variables in the Env-DATA system on Movebank, thus not shown in the table.

9	0.488	0.382	0.370	0.424	0.406	0.393	0.362	0	0.475	0.460	0.592
10	0.578	0.385	0.341	0.340	0.387	0.458	0.335	0.475	0	0.349	0.520
11	0.492	0.296	0.265	0.321	0.343	0.375	0.288	0.460	0.349	0	0.465
12	0.565	0.498	0.509	0.473	0.526	0.561	0.531	0.592	0.520	0.465	0

References

Agarwalla, A. K., & Minz, S. (2018). Unsupervised classification of remote sensing imagery using multi-sensor data fusion. *Proceedings of IEEE International Conference on Signal Processing and Communication, ICSPC 2017*, *2018-Janua*(July), 227–233. https://doi.org/10.1109/CSPC.2017.8305844

Aghabozorgi, S., Seyed Shirkhorshidi, A., & Ying Wah, T. (2015). Time-series clustering - A decade review. *Information Systems*, *53*, 16–38. https://doi.org/10.1016/j.is.2015.04.007

Ahearn, S. C., Dodge, S., Simcharoen, A., Xavier, G., & Smith, J. L. D. (2017). A context-sensitive correlated random walk: a new simulation model for movement. *International Journal of Geographical Information Science*, *31*(5), 867–883. https://doi.org/10.1080/13658816.2016.1224887

Bashir, F. I., Khokhar, A. A., & Schonfeld, D. (2007). Object trajectory-based activity classification and recognition using hidden Markov models. *IEEE Transactions on Image Processing*, *16*(7), 1912–1919. https://doi.org/10.1109/TIP.2007.898960

Belhadi, A., Djenouri, Y., Djenouri, D., Michalak, T., & Lin, J. C. W. (2021). Machine Learning for Identifying Group Trajectory Outliers. *ACM Transactions on Management Information Systems*, *12*(2), 1–25. https://doi.org/10.1145/3430195

Benkert, M., Gudmundsson, J., Hübner, F., & Wolle, T. (2008). *Reporting flock patterns*. *41*, 111–125. https://doi.org/10.1016/j.comgeo.2007.10.003

Berndt, D., & Clifford, J. (1994). Using dynamic time warping to find patterns in time series. *Workshop on Knowledge Knowledge Discovery in Databases*, *398*, 359–370. http://www.aaai.org/Papers/Workshops/1994/WS-94-03/WS94-03-031.pdf

Bildstein, K. L. (2006). *Migrating raptors of the world: their ecology and conservation*. Cornell University Press.

Bildstein, K. L., Barber, D., & Bechard, M. J. (2014). *Data from: Environmental drivers of variability in the movement ecology of turkey vultures (Cathartes aura) in North and South America*. Movebank data repository. https://doi.org/doi:10.5441/001/1.46ft1k05

Bildstein, K. L., Barber, D., Bechard, M. J., Graña Grilli, M., & Therrien, J. (2021). *Data from: Study "Vultures Acopian Center USA GPS" (2003-2021)*. Movebank data repository. https://doi.org/doi:10.5441/001/1.f3qt46r2

Bohrer, G., Brandes, D., Mandel, J. T., Bildstein, K. L., Miller, T. A., Lanzone, M., Katzner, T., Maisonneuve, C., & Tremblay, J. A. (2012). Estimating updraft velocity components over large spatial scales: Contrasting migration strategies of golden eagles and turkey vultures. *Ecology Letters*, *15*(2), 96–103. https://doi.org/10.1111/j.1461-0248.2011.01713.x

Bringmann, K., Künnemann, M., & Nusser, A. (2019). Walking the dog fast in practice: Algorithm engineering of the fréchet distance. *Leibniz International Proceedings in Informatics, LIPIcs*, *129*(17), 1–17. https://doi.org/10.4230/LIPIcs.SoCG.2019.17

Brum-Bastos, V., Long, J., & Demšar, U. (2016). Dynamic trajectory annotation for integrating environmental and movement data. *Visually-Supported Computational Movement Analysis Workshop - AGILE 2016*. http://viz.icaci.org/vcma2016/wp-content/uploads/2016/07/vcma_brumbastos-paper.pdf

Buchin, K., Buchin, M., & Gudmundsson, J. (2010). Constrained free space diagrams: A tool for trajectory analysis. *International Journal of Geographical Information Science*, *24*(7), 1101–1125. https://doi.org/10.1080/13658810903569598

Buchin, K., Buchin, M., van Leusden, R., Meulemans, W., & Mulzer, W. (2016). Computing the

Fréchet Distance with a Retractable Leash. *Discrete and Computational Geometry*, *56*(2), 315–336. https://doi.org/10.1007/s00454-016-9800-8

Buchin, M., & Wenk, C. (2020). Inferring movement patterns from geometric similarity. *Journal of Spatial Information Science*, *21*, 63–69. https://doi.org/10.5311/JOSIS.2020.21.724

Cheng, H., Liao, W., Yang, M. Y., Rosenhahn, B., & Sester, M. (2021). AMENet: Attentive Maps Encoder Network for trajectory prediction. *ISPRS Journal of Photogrammetry and Remote Sensing*, *172*(June 2020), 253–266. https://doi.org/10.1016/j.isprsjprs.2020.12.004

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1724–1734. https://doi.org/10.3115/v1/d14-1179

Choi, D., Min, K., & Choi, J. (2020). Regularising neural networks for future trajectory prediction via inverse reinforcement learning framework. *IET Computer Vision*, *14*(5), 192–200. https://doi.org/10.1049/iet-cvi.2019.0546

Choi, S., Kim, J., & Yeo, H. (2021). TrajGAIL: Generating urban vehicle trajectories using generative adversarial imitation learning. *Transportation Research Part C: Emerging Technologies*, *128*(May), 103091. https://doi.org/10.1016/j.trc.2021.103091

Cleasby, I. R., Wakefield, E. D., Morrissey, B. J., Bodey, T. W., Votier, S. C., Bearhop, S., & Hamer, K. C. (2019). Using time-series similarity measures to compare animal movement trajectories in ecology. *Behavioral Ecology and Sociobiology*, *73*(11). https://doi.org/10.1007/s00265-019-2761-1

Cyr, E. C., Gulian, M. A., Patel, R. G., Perego, M., & Trask, N. A. (2020). Robust Training and Initialization of Deep Neural Networks: An Adaptive Basis Viewpoint. *Proceedings of The First Mathematical and Scientific Machine Learning Conference, PMLR*, *107*, 512–536. http://arxiv.org/abs/1912.04862

Dabiri, S., & Heaslip, K. (2018). Inferring transportation modes from GPS trajectories using a convolutional neural network. *Transportation Research Part C: Emerging Technologies*, *86*, 360–371. https://doi.org/10.1016/j.trc.2017.11.021

Demchenko, Y., De Laat, C., & Membrey, P. (2014). Defining architecture components of the Big Data Ecosystem. *2014 International Conference on Collaboration Technologies and Systems, CTS 2014*, 104–112. https://doi.org/10.1109/CTS.2014.6867550

Djenouri, Y., Djenouri, D., & Lin, J. C. W. (2021). Trajectory outlier detection: New problems and solutions for smart cities. *ACM Transactions on Knowledge Discovery from Data*, *15*(2). https://doi.org/10.1145/3425867

Dodge, S., Bohrer, G., Bildstein, K., Davidson, S. C., Weinzierl, R., Bechard, M. J., Barber, D., Kays, R., Brandes, D., Han, J., & Wikelski, M. (2014). Environmental drivers of variability in the movement ecology of turkey vultures (cathartes aura) in North and South America. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *369*(1643). https://doi.org/10.1098/rstb.2013.0195

Dodge, S., Bohrer, G., Weinzierl, R., Davidson, S. C., Kays, R., Douglas, D., Cruz, S., Han, J., Brandes, D., & Wikelski, M. (2013). The Environmental-Data Automated Track Annotation (Env-DATA) system: linking animal tracks with environmental data to facilitate research of external factors effects on movement. *Movement Ecology*, *1*, 1–14.

Dodge, S., Laube, P., & Weibel, R. (2012). Movement similarity assessment using symbolic representation of trajectories. *International Journal of Geographical Information Science*, *26*(9), 1563–1588. https://doi.org/10.1080/13658816.2011.630003

Dos Santos, J. B., Heuser, C. A., Moreira, V. P., & Wives, L. K. (2011). Automatic threshold estimation for data matching applications. *Information Sciences*, *181*(13), 2685–2699. https://doi.org/10.1016/j.ins.2010.05.029

Eikelboom, J. A. J., de Knegt, H. J., Klaver, M., van Langevelde, F., van der Wal, T., & Prins, H. H. T. (2020). Inferring an animal's environment through biologging: quantifying the environmental influence on animal movement. *Movement Ecology*, *8*(1), 1–18. https://doi.org/10.1186/s40462-020-00228-4

Eler, D. M., Teixeira, J. B. M., Macanha, P. A., & Garcia, R. E. (2015). Simplified stress and simplified silhouette coefficient to a faster quality evaluation of multidimensional projection techniques and feature spaces. *Proceedings of the International Conference on Information Visualisation*, *2015-Septe*, 133–139. https://doi.org/10.1109/iV.2015.33

Endo, Y., Toda, H., Nishida, K., & Ikedo, J. (2016). Classifying spatial trajectories using representation learning. In *International Journal of Data Science and Analytics* (Vol. 2, Issues 3–4, pp. 107–117). https://doi.org/10.1007/s41060-016-0014-1

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, PMLR*, *9*, 249–256.

Grant, T. J., Parry, H. R., Zalucki, M. P., & Bradbury, S. P. (2018). Predicting monarch butterfly (Danaus plexippus) movement and egg-laying with a spatially-explicit agent-based model: The role of monarch perceptual range and spatial memory. *Ecological Modelling*, *374*(February), 37–50. https://doi.org/10.1016/j.ecolmodel.2018.02.011

Gudmundsson, J., & Valladares, N. (2015). A GPU approach to subtrajectory clustering using the Fréchet distance. *IEEE Transactions on Parallel and Distributed Systems*, *26*(4), 924–937. https://doi.org/10.1109/TPDS.2014.2317713

Guo, S., Mou, J., Chen, L., & Chen, P. (2021). Improved kinematic interpolation for AIS trajectory reconstruction. *Ocean Engineering*, *234*(May), 109256. https://doi.org/10.1016/j.oceaneng.2021.109256

Guo, Y., B, Q. X., Fan, Y., Liang, S., & Sbert, M. (2016). *Fast Agglomerative Information Bottleneck Based Trajectory Clustering*. *1*, 425–433. https://doi.org/10.1007/978-3-319-46675-0

Gupta, A., Johnson, J., Silvio, L. F., & Alexandre, S. (2018). Social GAN : Socially Acceptable Trajectories with Generative Adversarial Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2255–2264.

Hägerstrand, T. (1970). What about people in Regional Science? *Papers of the Regional Science Association*, *24*(1), 6–21. https://doi.org/10.1007/BF01936872

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I. W., & Sugiyama, M. (2018). Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in Neural Information Processing Systems*, 8527–8537.

Hirakawa, T., Yamashita, T., Tamaki, T., Fujiyoshi, H., Umezu, Y., Takeuchi, I., Matsumoto, S., & Yoda, K. (2018). Can AI predict animal movements? Filling gaps in animal trajectories using inverse reinforcement learning. *Ecosphere*, *9*(10). https://doi.org/10.1002/ecs2.2447

Ho, T. K. (1995). Random decision forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995*, 278–282.

Hochreiter, S., & Urgen Schmidhuber, J. (1997). Long Shortterm Memory. *Neural Computation*, *9*(8), 17351780.

Jäger, G. (2019). Replacing rules by neural networks a framework for agent-based modelling. *Big Data and Cognitive Computing*, *3*(4), 1–12. https://doi.org/10.3390/bdcc3040051

Juarez, C., Messina, A. R., Castellanos, R., & Espinosa-Pérez, G. (2011). Characterization of multimachine system behavior using a hierarchical trajectory cluster analysis. *IEEE Transactions on Power Systems*, *26*(3), 972–981. https://doi.org/10.1109/TPWRS.2010.2100051

Khoshaein, V. (2013). *Trajectory Clustering using a Variation of Fréchet Distance*.

Kingma, D. P., & Ba, J. L. (2015, December 22). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. https://arxiv.org/abs/1412.6980v9

Kolendo, P., & Śmierzchalski, R. (2016). Experimental Comparison of Straight Lines and Polynomial Interpolation Modeling Methods in Ship Evolutionary Trajectory Planning Problem.pdf. *Advanced and Intelligent Computations in Diagnosis and Control*, *386*, 331–340. https://doi.org/10.1007/978-3-319-23180-8_24

Konzack, M., McKetterick, T., Ophelders, T., Buchin, M., Giuggioli, L., Long, J., Nelson, T., Westenberg, M. A., & Buchin, K. (2017). Visual analytics of delays and interaction in movement data. *International Journal of Geographical Information Science*, *31*(2), 320–345. https://doi.org/10.1080/13658816.2016.1199806

Kranstauber, B. (2019). Modelling animal movement as Brownian bridges with covariates. *Movement Ecology*, *7*(1), 1–10. https://doi.org/10.1186/s40462-019-0167-3

Kranstauber, B., Cameron, A., Weinzerl, R., Fountain, T., Tilak, S., Wikelski, M., & Kays, R. (2011). The Movebank data model for animal tracking. *Environmental Modelling and Software*, *26*(6), 834–835. https://doi.org/10.1016/j.envsoft.2010.12.005

Lan, T., Hu, H., Jiang, C., Yang, G., & Zhao, Z. (2020). A comparative study of decision tree , random forest , and convolutional neural network for spread-F identification. *Advances in Space Research*, *65*(8), 2052–2061. https://doi.org/10.1016/j.asr.2020.01.036

Laube, P. (2014). Computational movement analysis. In *Springer Handbook of Geographic Information*. https://doi.org/10.1007/978-3-540-72680-7_22

Laube, P., Dennis, T., Forer, P., & Walker, M. (2007). Movement beyond the snapshot - Dynamic analysis of geospatial lifelines. *Computers, Environment and Urban Systems*, *31*(5), 481–501. https://doi.org/10.1016/j.compenvurbsys.2007.08.002

Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. In *Soviet Physics Doklady* (Vol. 10, Issue 8, pp. 707–710). https://doi.org/10.1016/S0074-7742(08)60036-7

Li, W., Zhang, C., Ma, J., & Jia, C. (2019). Long-term vessel motion predication by modeling trajectory patterns with AIS data. *ICTIS 2019 - 5th International Conference on Transportation Information and Safety*, 1389–1394. https://doi.org/10.1109/ICTIS.2019.8883596

Li, Yali, Xiang, L., Zhang, C., & Wu, H. (2019). Fusing taxi trajectories and rs images to build road map via dcnn. *IEEE Access*, *7*. https://doi.org/10.1109/ACCESS.2019.2951730

Li, Yingmin, Chen, H., & Wu, Z. (2010). Dynamic time warping distance method for similarity test of multipoint ground motion field. *Mathematical Problems in Engineering*, *2010*. https://doi.org/10.1155/2010/749517

Liu, F., Bi, W., Hao, W., Gao, F., & Tang, J. (2021). An Improved Fuzzy Trajectory Clustering Method for Exploring Urban Travel Patterns. *Journal of Advanced Transportation*, *2021*(3). https://doi.org/10.1155/2021/6651718

Liu, H., Li, J., Wu, Y., & Fu, Y. (2019). Clustering With Outlier Removal. *IEEE Transactions on Knowledge and Data Engineering*, *33*(6), 2369–2379. https://doi.org/10.1109/tkde.2019.2954317

Long, J. A. (2016). Kinematic interpolation of movement data Kinematic interpolation of movement data. *International Journal of Geographical Information Science*, *30*(5), 854–868. https://doi.org/10.1080/13658816.2015.1081909

Mallon, J. M., Bildstein, K. L., & Fagan, W. F. (2021). Inclement weather forces stopovers and prevents migratory progress for obligate soaring migrants. *Movement Ecology*, *9*(1), 1–9. https://doi.org/10.1186/s40462-021-00274-6

Mao, J., Liu, J., Jin, C., & Zhou, A. (2021). Feature Grouping-based Trajectory Outlier Detection over Distributed Streams. *ACM Transactions on Intelligent Systems and Technology*, *12*(2). https://doi.org/10.1145/3444753

Mehri, S., Alesheikh, A. A., & Basiri, A. (2021). A Contextual Hybrid Model for Vessel Movement Prediction. *IEEE Access*, *9*, 45600–45613. https://doi.org/10.1109/ACCESS.2021.3066463

Meng, F., Yuan, G., Lv, S., Wang, Z., & Xia, S. (2019). An overview on trajectory outlier detection. *Artificial Intelligence Review*, *52*(4), 2437–2456. https://doi.org/10.1007/s10462-018-9619-1

Michelot, T., Langrock, R., & Patterson, T. A. (2016). moveHMM : an R package for the statistical modelling of animal movement data using hidden Markov models. *Methods in Ecology and Evolution*, *7*, 1308–1315. https://doi.org/10.1111/2041-210X.12578

Miller, H. J. (2005). A measurement theory for time geography. *Geographical Analysis*, *37*(1), 17–45. https://doi.org/10.1111/j.1538-4632.2005.00575.x

Miller, H. J. (2008). Geographic Data Mining and Knowledge Discovery. In *The Handbook of Geographic Information Science* (pp. 352–366). John Wiley and Sons. https://doi.org/10.1002/9780470690819.ch19

Moayedi, A., Abbaspour, R. A., & Chehreghan, A. (2019). An evaluation of the efficiency of similarity functions in density-based clustering of spatial trajectories. *Annals of GIS*, *25*(4), 313–327. https://doi.org/10.1080/19475683.2019.1679254

Moseley, B., & Wang, J. R. (2017). Approximation bounds for hierarchical clustering: Average linkage, bisecting K-means, and Local Search. *Advances in Neural Information Processing Systems*, *2017-Decem*(Nips), 3095–3104.

Nathan, R., Getz, W. M., Revilla, E., Holyoak, M., Kadmon, R., Saltz, D., & Smouse, P. E. (2008). *A movement ecology paradigm for unifying organismal movement research*. *105*(49), 19052–19059.

Nawaz, A., Zhiqiu, H., Senzhang, W., Hussain, Y., Khan, I., & Khan, Z. (2020). Convolutional LSTM based transportation mode learning from raw GPS trajectories. *IET Intelligent Transport Systems*, *14*(6), 570–577. https://doi.org/10.1049/iet-its.2019.0017

Parent, C., Spaccapietra, S., Renso, C., Andrienko, G., Andrienko, N., Bogorny, V., Damiani, M. L., Macedo, J., Pelekis, N., Theodoridis, Y., & Yan, Z. (2013). Semantic Trajectories Modeling and Analysis. *ACM Computing Surveys*, *45*(4), 1–32.

Park, C. H. (2019). Outlier and anomaly pattern detection on data streams. *Journal of Supercomputing*, *75*(9), 6118–6128. https://doi.org/10.1007/s11227-018-2674-1

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*(NeurIPS), 8024–8035.

Patterson, T. A., Basson, M., Bravington, M. V., & Gunn, J. S. (2009). Classifying movement behaviour in relation to environmental conditions using hidden Markov models. *Journal of Animal Ecology*, *78*(6), 1113–1123. https://doi.org/10.1111/j.1365-2656.2009.01583.x

Patterson, T. A., Thomas, L., Wilcox, C., Ovaskainen, O., & Matthiopoulos, J. (2008). State – space models of individual animal movement. *Trends in Ecology & Evolution*, *23*(2), 87–94. https://doi.org/10.1016/j.tree.2007.10.009

Qin, W., Tang, J., Lu, C., & Lao, S. (2021). *Trajectory prediction based on long short-term memory network and Kalman filter using hurricanes as an example*. 1005–1023.

Quinlan, J. r. (1986). Induction of Decision Trees. *Machine Learning*, *1*, 81–106. https://doi.org/10.1023/A:1022643204877

Ranacher, P., & Tzavella, K. (2014). How to compare movement? A review of physical movement similarity measures in geographic information science and beyond. *Cartography and Geographic Information Science*, *41*(3), 286–307. https://doi.org/10.1080/15230406.2014.890071

Reddi, S. J., Kale, S., & Kumar, S. (2018). On the convergence of Adam and beyond. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 1–23.

Ren, C., Tang, L., Long, J., Kan, Z., & Yang, X. (2021). Modelling place visit probability sequences during trajectory data gaps based on movement history. *ISPRS International Journal of Geo-Information*, *10*(7). https://doi.org/10.3390/ijgi10070456

Rew, J., Park, S., Cho, Y., Jung, S., & Hwang, E. (2019). Animal movement prediction based on predictive recurrent neural network. *Sensors (Switzerland)*, *19*(20), 4411. https://doi.org/10.3390/s19204411

Rick, C. (2000). *Efficient Computation of All Longest Common Subsequences*. 407–418. https://doi.org/10.1007/3-540-44985-x_35

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*(C), 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

Rowcliffe, J. M., Carbone, C., Kays, R., Kranstauber, B., & Jansen, P. A. (2012). Bias in estimating animal travel distance: The effect of sampling frequency. *Methods in Ecology and Evolution*, *3*(4), 653–662. https://doi.org/10.1111/j.2041-210X.2012.00197.x

Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H., & Savarese, S. (2019). SoPhie: An attentive GAN for predicting paths compliant to social and physical constraints. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2019-June*, 1349–1358. https://doi.org/10.1109/CVPR.2019.00144

Seideman, J. D., Khan, B., & Vargas, A. C. (2014). Identifying malware genera using the Jensen-Shannon distance between system call traces. *Proceedings of the 9th IEEE International Conference on Malicious and Unwanted Software, MALCON 2014*, 1–7. https://doi.org/10.1109/MALWARE.2014.6999409

Shamoun-Baranes, J., Bouten, W., & Van Loon, E. E. (2010). Integrating meteorology into research on migration. *Integrative and Comparative Biology*, *50*(3), 280–292. https://doi.org/10.1093/icb/icq011

Sharif, M., & Alesheikh, A. A. (2017). Context-awareness in similarity measures and pattern discoveries of trajectories: a context-based dynamic time warping method. *GIScience and*

Remote Sensing, *54*(3), 426–452. https://doi.org/10.1080/15481603.2017.1278644

Shi, X., Shao, X., Guo, Z., Wu, G., Zhang, H., & Shibasaki, R. (2019). Pedestrian trajectory prediction in extremely crowded scenarios. *Sensors (Switzerland)*, *19*(5), 1–18. https://doi.org/10.3390/s19051223

Siła-Nowicka, K., Vandrol, J., Oshan, T., Long, J. A., Demšar, U., & Fotheringham, A. S. (2016). Analysis of human mobility patterns from GPS trajectories and contextual information. *International Journal of Geographical Information Science*, *30*(5), 881–906. https://doi.org/10.1080/13658816.2015.1100731

Spiegel, O., Leu, S. T., Bull, C. M., & Sih, A. (2017). What ' s your move ? Movement as a link between personality and spatial dynamics in animal populations. *Ecology Letters*, *20*, 3–18. https://doi.org/10.1111/ele.12708

Squires, J. R., DeCesare, N. J., Olson, L. E., Kolbe, J. A., Hebblewhite, M., & Parks, S. A. (2013). Combining resource selection and movement behavior to predict corridors for Canada lynx at their southern range periphery. *Biological Conservation*, *157*, 187–195. https://doi.org/10.1016/j.biocon.2012.07.018

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Su, H., Liu, S., Zheng, B., Zhou, X., & Zheng, K. (2020). A survey of trajectory distance measures and performance evaluation. *VLDB Journal*, *29*(1), 3–32. https://doi.org/10.1007/s00778-019-00574-9

Sun, Z., Chen, Y., Wang, P., Fang, S., & Tang, B. (2021). Vision-Based Traffic Conflict Detection Using Trajectory Learning and Prediction. *IEEE Access*, *9*, 34558–34569. https://doi.org/10.1109/ACCESS.2021.3061266

Tao, Y., Both, A., Silveira, R. I., Buchin, K., Sijben, S., Purves, R. S., Laube, P., Peng, D., Toohey, K., & Duckham, M. (2021). A comparative analysis of trajectory similarity measures. *GIScience and Remote Sensing*, *58*(5), 643–669. https://doi.org/10.1080/15481603.2021.1908927

Technitis, G., Othman, W., Safi, K., & Weibel, R. (2015). From A to B, randomly: a point-to-point random trajectory generator for animal movement. *International Journal of Geographical Information Science*, *29*(6), 912–934. https://doi.org/10.1080/13658816.2014.999682

Toohey, K., & Duckham, M. (2015). Trajectory similarity measures. *SIGSPATIAL Special*, *7*(1), 43–50. https://doi.org/10.1145/2782759.2782767

Tremblay, Y., Shaffer, S. A., Fowler, S. L., Kuhn, C. E., Mcdonald, B. I., Weise, M. J., Bost, C., Weimerskirch, H., Crocker, D. E., Goebel, M. E., & Costa, D. P. (2006). *Interpolation of animal tracking data in a fluid environment*. *209*, 128–140. https://doi.org/10.1242/jeb.01970

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … Vázquez-Baeza, Y. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2

Vlachos, M., Gunopulos, D., & Das, G. (2004). Rotation invariant distance measures for trajectories. *KDD-2004 - Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 707–712. https://doi.org/10.1145/1014052.1014144

Vlachos, M., Kollios, G., & Gunopulos, D. (2002). Discovering similar multidimensional trajectories. *Proceedings - International Conference on Data Engineering*, 673–684. https://doi.org/10.1109/ICDE.2002.994784

Wang, C., Zourlidou, S., Golze, J., & Sester, M. (2021). Trajectory analysis at intersections for traffic rule identification. *Geo-Spatial Information Science*, *24*(1), 75–84. https://doi.org/10.1080/10095020.2020.1843374

Wang, H., Su, H., Zheng, K., Sadiq, S., & Zhou, X. (2013). An effectiveness study on trajectory similarity measures. *Conferences in Research and Practice in Information Technology Series*, *137*(February), 13–22.

Wentz, E. A., Campbell, A. F., Houston, R., Wentz, E. A., Campbell, A. F., & A, R. H. (2003). *A comparison of two methods to create tracks of moving objects : linear weighted distance and constrained random walk*. *17*(7), 623–645. https://doi.org/10.1080/1365881031000135492

Winter, S., & Yin, Z. (2010). *Directed movements in probabilistic time geography*. *24*(9), 1349–1365. https://doi.org/10.1080/13658811003619150

Xiao, Z., Wang, Y., Fu, K., & Wu, F. (2017). Identifying different transportation modes from trajectory data using tree-based ensemble classifiers. *ISPRS International Journal of Geo-Information*, *6*(2). https://doi.org/10.3390/ijgi6020057

Xu, J., Sun, X., Zhang, Z., Zhao, G., & Lin, J. (2019). Understanding and improving layer normalization. *Advances in Neural Information Processing Systems*, *32*, 1–11.

Yang, X., Hou, L., Guo, M., Cao, Y., Yang, M., & Tang, L. (2022). Road intersection identification from crowdsourced big trace data using Mask-RCNN. *Transactions in GIS*, *26*(1), 278–296. https://doi.org/10.1111/tgis.12851

Yu, B., & Kim, S. H. (2006). Interpolating and Using Most Likely Trajectories in Moving-Objects Databases. *Database and Expert Systems Applications. DEXA 2006. Lecture Notes in Computer Science, Vol 4080. Springer, Berlin, Heidelberg.*, 718–727. https://doi.org/10.1007/11827405_70

Zhang, C., Bin, J., Wang, W., Peng, X., Wang, R., Halldearn, R., & Liu, Z. (2020). AIS data driven general vessel destination prediction: A random forest based approach. *Transportation Research Part C: Emerging Technologies*, *118*(April 2019), 102729. https://doi.org/10.1016/j.trc.2020.102729

Zheng, Y., Capra, L., Wolfson, O., & Yang, H. (2014). Urban Computing: Concepts, Methodologies, and Applications. *ACM Transactions on Intelligent Systems and Technology*, *5*(3), 1–55. https://doi.org/10.1145/2629592

Zhu, J., Cheng, D., Zhang, W., Song, C., Chen, J., & Pei, T. (2021). A New Approach to Measuring the Similarity of Indoor Semantic Trajectories. *ISPRS International Journal of Geo-Information*, *10*(2), 90. https://doi.org/10.3390/ijgi10020090