A Generative Trajectory Interpolation Method for Imputing Gaps in Wildlife Movement Data

Zijian Wan
zijian.wan@geog.ucsb.edu
University of California, Santa Barbara
Santa Barbara, California, USA

Somayeh Dodge
University of California, Santa Barbara
Santa Barbara, California, USA
sdodge@geog.ucsb.edu

ABSTRACT

Advances in tracking technologies have resulted in growing repositories of large and long-term movement data of wildlife at an unprecedented rate. Nevertheless, many of these movement datasets come with missing records, termed *gaps* in this paper, which need to be imputed before further movement analysis. However, existing trajectory interpolation methods have certain limitations. Their effectiveness might be restrained by users' domain knowledge of the moving entity or by the properties of the trajectories, to name a few. Moreover, the uncertainty of movement data has not received enough attention and is often neglected in the interpolation process. A review of existing literature suggests a need for designing more robust and broadly applicable data-driven interpolation methods that can self-adapt to the subject tracking data, and meanwhile, can take movement uncertainty into consideration. This study proposes a new trajectory interpolation model that leverages a generative adversarial network (GAN) architecture supported by long short-term memory (LSTM) layers to interpolate missing trajectory points. The model uses a latent code in addition to the noise input to deal with the uncertainty in movement behaviors. We apply and evaluate the proposed model against a real-world GPS trajectory dataset of migratory white storks to assess its effectiveness for imputing migration paths.

CCS CONCEPTS

• **Computing methodologies** → **Modeling methodologies**; *Neural networks.*

KEYWORDS

Trajectory interpolation, movement modeling, uncertainty, generative adversarial network (GAN), long short-term memory (LSTM), GPS trajectories, wildlife tracking

ACM Reference Format:
Zijian Wan and Somayeh Dodge. 2023. A Generative Trajectory Interpolation Method for Imputing Gaps in Wildlife Movement Data. In *1st ACM SIGSPATIAL International Workshop on AI-driven Spatio-temporal Data Analysis for Wildlife Conservation (GeoWildLife '23), November 13, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3615893.3628759

GeoWildLife '23, November 13, 2023, Hamburg, Germany

ACM ISBN 979-8-4007-0355-3/23/11.
https://doi.org/10.1145/3615893.3628759

1 INTRODUCTION

Novel tracking devices and technologies (e.g., Global Positioning System (GPS) tracking collars) are providing scientists and wildlife conservationists with wildlife tracking data at an unprecedented speed and at much lower costs. These tracking devices usually sample the location of wildlife at mostly regular intervals, which forms sequences of timestamped locations, named trajectories. Computational movement ecology and other wildlife conservation analyses are developed on the foundation of reliable tracking data [17]. Nevertheless, many factors (e.g., battery outage, signal loss, signal multi-pass) may lead to the interruption of tracking data recording or introducing erroneous data points and outliers that need to be considered before further analysis. This results in missing data and often long gaps in movement data. In this study, a sequence of consecutive missing tracking points is termed as a *gap*. It originates from the scenario where the tracked animal continues moving, but the tracking device stops tracking or when the recording is interrupted by the environmental and mechanical conditions surrounding the sensors. These forms of gaps often create a significant problem since they may appear stochastically. Gaps may also be intentionally introduced into the dataset. For instance, some animal trackers are solar-powered and thus are designed to turn off automatically after sunset to save battery, which leads to irregular sampling rates. However, there is a special case, although not the focus of this paper, where some trackers may pause recording when an animal is inactive. In this case, the missing points are merely temporal but not spatial, as the moving entity remains at the same location. For a more comprehensive understanding and representation of movement and its patterns through data-driven approaches, gaps are the primary targets that need to be dealt with before further analysis.

In movement ecology, there is often a desire and necessity to estimate a moving entity's unknown locations according to the observed ones. The process of actively filling the gap by estimating the locations (and other attributes if necessary) of the missing points along a trajectory is termed *trajectory interpolation*. As shown in Figure 1, for a trajectory containing a gap, the goal of trajectory interpolation is to obtain the estimation of unknown locations (denoted as $\hat{y}$), given two observed segments at two ends of that trajectory (denoted as x_1, x_2). A review of existing literature suggests a need for designing more robust and broadly applicable data-driven interpolation methods that can self-adapt to the subject tracking data [7, 15, 30, 27]. Meanwhile, another aspect that is understudied in trajectory interpolation is the uncertainty of movement path choices. As shown in Figure 1, there might be multiple possible and acceptable paths in the gap that a moving entity might choose, and

thus a robust interpolation method needs to take uncertainty into consideration.

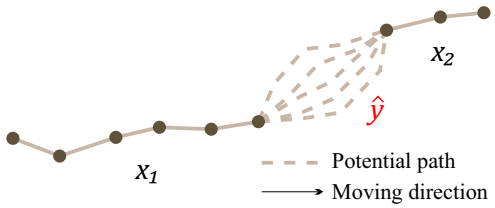

Figure 1: Problem definition of trajectory interpolation

To bridge this research gap, this paper aims to propose a generic generative trajectory interpolation model that can unravel the knowledge needed to interpolate a gap from the observed tracking data, especially from the very trajectory that contains the gap. The proposed model relies on a generative adversarial network (GAN) supported by long short-term memory (LSTM). The main contributions of this work are as follows:

(1) A generative adversarial network (GAN) supported by long short-term memory (LSTM) is proposed to impute gaps contained in GPS trajectories in the vector space. We fuse the two pieces of information learned from two ends of a trajectory containing a gap to form the interpolation context for each individual trajectory, guiding the estimation of unknown locations. This makes the proposed model distinct from existing models using a similar structure for trajectory prediction (extrapolation) in the raster (pixelated) space.
(2) Based on InfoGAN [3], this study uses a latent code in addition to the noise input, enabling the proposed model to generate diverse interpolation results, which provides a better approximation to the gap ground truth (real trajectory) and helps deal with the inherent uncertainty in movement path choices.

2 RELATED WORK

2.1 Trajectory interpolation

There are many existing trajectory interpolation techniques put forward to deal with gaps in movement data. A classic method is linear interpolation, which assumes that the entity is moving at a constant speed and heading in the interpolated area. The most important supremacy of linear interpolation is that it can be easily and straightforwardly implemented and requires little computational resources, which might make a substantial difference when a movement dataset is in tremendous volume. However, for a large gap or movement involving more complicated patterns, the trajectory segment produced from linear interpolation is often oversimplified, especially when the temporal resolution of the data is coarse. One important reason is that some extent of stochasticity exists in movement. Thus, some researchers model movement as a probabilistic random process, e.g., random walks [24, 26], or with uncertainty, using Brownian bridges [12, 10] or a potential path area (PPA) [1, 16]. These approaches are suitable for many types of animals, especially terrestrial animals since they manifest more random movement patterns. Nevertheless, the major challenge is to precisely parameterize the probabilistic rules, which often requires sufficient domain knowledge of the moving process, including knowledge of the moving entities and the surrounding environment. Consequently, movement models using random walks, Brownian bridges, and PPA can generate a general movement coverage area over a long period of time, but it is usually challenging for them to estimate precisely where a moving entity is at a given timestamp.

To solve that problem, researchers turn to movement modeling with mathematical and statistical support. 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 [30]. In this way, the location of a moving entity at any given timestamp can be estimated since the whole movement process is modeled mathematically. Trajectory interpolation supported by curve fitting is proven suitable for some marine mammals as their movement does demonstrate regular geometric patterns [27]. However, such interpolation methods focus on the spatial features only and neglect the temporal or spatiotemporal features implied in trajectories, such as time and speed. To incorporate spatiotemporal features, some researchers put forward the kinematic interpolation methods [7, 15]. This kind of interpolation method usually includes a series of carefully designed complex kinematic functions, which can result in high accuracy if one has sufficient domain knowledge to tune the parameters. Unfortunately, that characteristic also restricts the robustness and adaptiveness of these kinematic interpolation methods. When the set of kinematic functions is tuned for one movement dataset, it usually cannot be applied to other types of moving entities and sometimes even different behavior patterns manifested by the same entity might cause a disturbance to the interpolation performance. Therefore, it is needed to design a data-driven trajectory interpolation method that can self-adapt to movement datasets and more importantly, to the very trajectory containing the gap.

2.2 Machine learning approaches to model movement trajectories

Recent years have seen the versatility and effectiveness of machine learning proven in an increasing number of domains and fields. Through sufficient training, deep neural networks can capture and unravel high-level features of the trajectories and identify implicit patterns that are otherwise imperceptible [13, 28]. Long short-term memory (LSTM), a special kind of recurrent neural network (RNN), stimulates the interests of movement researchers since its structure makes it inherently suitable for processing time series data [9]. As an improved version of the traditional RNN, LSTM has the capability to apprehend both long-term and short-term dependencies, which are key to making a well-informed estimation of unknown timesteps in sequential data [5]. To further improve the learning and prediction capabilities, LSTMs are used inside the architecture of a generative adversarial network (GAN) to predict pedestrian trajectories using pixel-based pedestrian tracking datasets captured by cameras. A GAN is an architecture for training deep generative models based on a minimax game that is put formally in Equation

(1) [6].

$$\min_G \max_D V_{\text{GAN}}(D,G) = \mathbb{E}_{x\sim P_{\text{data}}}[\log D(x)] + \mathbb{E}_{z\sim P_{\text{noise}}}\{1-\log[1-D(G(z))]\} \quad (1)$$

It has been proven in a variety of studies that the GAN architecture can lead to a considerable increase in model performance [2, 8, 25]. One major limitation of traditional GANs, however, is their invariability, and thus they cannot deal with the uncertainty in movement modeling [20]. In the trajectory interpolation context, given a trajectory containing a gap, an interpolation model based on the traditional GAN will only generate one interpolation result that it believes to be the most probable. Even if asked to do the interpolation task multiple times, the model will generate results that are highly similar. This is because apart from the input trajectory, the only input this model takes is a random noise z, of which we have little control. Hence, its outputs cannot cover diverse possibilities, which might exist in reality because of movement and behavior uncertainty. A variant of traditional GANs, named InfoGAN [3], is proposed to solve that problem. In addition to the random noise input z, it adds a latent code c to model the uncertainty of movement behaviors. The introduction of such a latent code not only improves the model's capability of dealing with uncertainty but also helps avoid the mode collapsing problem, which is common in the training process of traditional [2]. With different inputs of c, an InfoGAN-based model varies its prediction, accordingly, making it feasible to create diverse interpolation results among all possibilities. Note that this association between the latent code and movement behavior patterns is also learned from trajectories. To put it formally, an InfoGAN solves the information-regularized minimax game shown in Equation (2).

$$\min_{G,Q}\max_D V_{\text{InfoGAN}}(D,G,Q) = V(D,G) - \lambda L_I(G,Q) \quad (2)$$

where λ is a hyperparameter, $Q(c|x)$ is an auxiliary distribution, and $L_I(G,Q)$ is a variational lower bound of the mutual information $I(c;G(z,c))$. For more detailed information on how that is derived, refer to [3].

The InfoGAN has stimulated the interest of researchers studying movement (e.g., predicting pedestrian trajectories [2]), for its ability to generate diverse results by varying the latent code c. Inspired by previous work, this study aims to develop an InfoGAN-based trajectory interpolation model, which differs from previous works in the following aspects. First, this study focuses on trajectory interpolation while most previous works, especially the ones built upon a similar architecture, focus on prediction (extrapolation). Second, many of those models are developed in the field of computer vision, and thus the trajectories considered in those models are usually quite short (e.g., lasting for less than 20 timesteps with the observed and predicted segments combined) and the sampling rate is usually very high. In videos, moving entities are often sampled at the rate of milliseconds. Although in many studies, such video movement datasets are down-sampled to reduce the computational resources needed, they are still considerably high compared to the sampling rate of GPS trajectories collected for wildlife. Third, they work on rasterized trajectories which may compromise the accuracy of movement locations. Therefore, this study advances the LSTM-GAN to develop a trajectory interpolation model applicable to wildlife GPS trajectories, which have longer durations and lower sampling rates, compared to trajectories captured in videos.

3 LSTM-GAN TRAJECTORY INTERPOLATION MODEL

The proposed generative trajectory interpolation model consists of a long short-term memory (LSTM) encoder-decoder generator and an LSTM discriminator. When interpolating a gap, the generator first encodes the observed segments of that very trajectory sequentially to learn individual patterns and then decodes the gap with the individual movement pattern as the prior knowledge. The discriminator supervises the generator's work, making certain the interpolation result is close to the ground truth. Figure 2 demonstrates a schematic illustration of the proposed interpolation model with the GAN architecture that can transform a trajectory with a gap (see Figure 2b) to a complete trajectory with the gap interpolated (see Figure 2c).

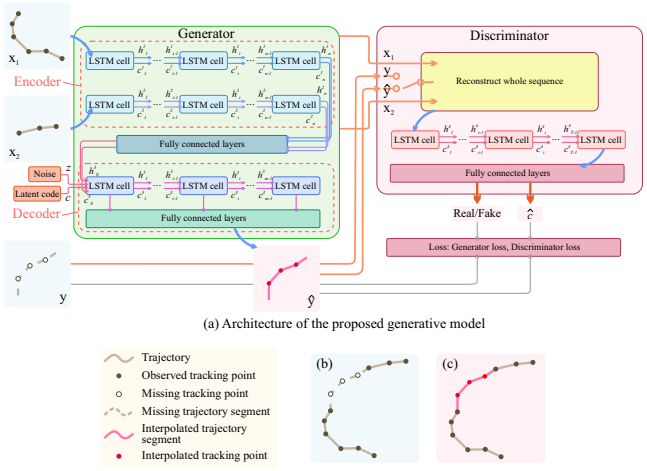

Figure 2: Schematic illustration of the proposed trajectory interpolation model

3.1 Generator

In this study, the generator has an LSTM encoder-decoder structure, enabling the generator to read the whole trajectory sequence before estimating the unknown locations in the gap. The encoder first obtains knowledge about movement patterns from the observed segments (denotes as x_1 and x_2) at two ends of a trajectory containing a gap (see Figure 1 for the illustration of trajectory interpolation problem definition). Here, a trajectory is represented as a $n \times k$ matrix, where n denotes the total time step and k denotes the number of attributes used to represent movement (e.g., location, speed, heading). For an input trajectory, the encoder LSTM cells encode the observed segments x_1 and x_2 to obtain the hidden states h^t_{enc1} and h^t_{enc2} at time step t through Equations (3-4)

$$h^t_{\text{enc1}} = \lambda_{\text{enc1}}(h^{t-1}_{\text{enc1}}, x^t_1; W_{\lambda_{\text{enc1}}}) \quad (3)$$

$$h^t_{\text{enc2}} = \lambda_{\text{enc2}}(h^{t-1}_{\text{enc2}}, x^t_2; W_{\lambda_{\text{enc2}}}) \quad (4)$$

where $\lambda(\cdot)$ represents an encoder LSTM with weight W.

The two pieces of information learned from these two segments are then fused through fully connected layers to obtain the final hidden state of the encoder h^n_{enc} through Equation (5).

$$h^n_{\text{enc}} = \phi_{\text{enc}}(h^n_{\text{enc1}}, h^n_{\text{enc2}}; W_{\phi_{\text{enc}}}) \tag{5}$$

where $\phi_{\text{enc}}(\cdot)$ represents the fully connected layers with the *LeakyReLu* activation function and weight $W_{\phi_{\text{enc}}}$.

Since our goal is to generate gap interpolations that are consistent with what has been observed, the hidden state of the decoder LSTM is initialized as $h_{\text{dec}^0} = h^n_{\text{enc}}$. After initializing h_{dec^0}, the decoder interpolates the gap (denoted as $\hat{y}$) using what is learned by the encoders as prior knowledge through the recurrence of Equations (6-8).

$$o^{t-1} = [\hat{y}^{t-1}, z, c] \tag{6}$$

$$h^t_{\text{dec}} = \lambda_{\text{dec}}(h^{t-1}_{\text{dec}}, o^{t-1}; W_{\lambda_{\text{dec}}}) \tag{7}$$

$$\hat{y}^t = \phi_{\text{dec}}(h^t_{\text{dec}}; W_{\phi_{\text{dec}}}) \tag{8}$$

where $z \sim \mathcal{N}(0, 1)$ denotes noise, c is a latent code, $[\cdot]$ represents concatenation, o^{t-1} denotes the input of the decoder LSTM, $\hat{y}^t$ denotes the output of the decoder fully connected layers $\phi_{\text{dec}}(\cdot)$ with weight $W_{\phi_{\text{dec}}}$, and h_{dec^t} is the hidden state of the decoder LSTM $\lambda_{\text{dec}}(\cdot)$ at time step t.

3.2 Discriminator

In the discriminator module, we first reconstruct the complete trajectory. The discriminator has an equal chance of selecting the ground truth of the gap (denoted as y) or the interpolation result output by the generator (denoted as $\hat{y}$). In this way, the probabilities are equal that the discriminator sees a real trajectory $T_1 = [x_1, y, x_2]$ or a fake trajectory $T_0 = [x_1, \hat{y}, x_2]$. Suppose we denote a reconstructed trajectory (either real or fake) as $T = \{p^1, p^2, \ldots, p^n\}$, where p^i represents the tracking point at time step i. It is processed by the discriminator LSTM layer to obtain the hidden state h^t_{disc} at time step t through Equation (9).

$$h^t_{\text{disc}} = \lambda_{\text{disc}}(h^{t-1}_{\text{disc}}, p^t; W_{\lambda_{\text{disc}}}) \tag{9}$$

where $\lambda(\cdot)$ represents an encoder LSTM with weight W.

After processing the whole sequence, we obtain the final hidden state h^n_{disc}, encoding the important information extracted from trajectory T. Based on this information, the discriminator then differentiates whether it sees a real or fake trajectory through Equation (10) and then decodes the latent code $\hat{c}$ through Equation (11).

$$\hat{l} = \phi_{\text{label}}(h^n_{\text{disc}}; W_{\phi_{\text{label}}}) \tag{10}$$

$$\hat{c} = \phi_c(h^n_{\text{disc}}; W_{\phi_c}) \tag{11}$$

where $\phi_{\text{label}}(\cdot)$ represents the fully connected layers with the *Sigmoid* activation function and weight $W_{\phi_{\text{label}}}$, and $\phi_c(\cdot)$ represents the fully connected layers with the *LeakyReLu* activation function and weight W_{ϕ_c}. $\hat{l}$ and $\hat{c}$ are the label (real or fake) and latent code predicted by the discriminator.

In such context, the terms $V(D, G)$ and $L_I(G, Q)$ in Equation (2) are instantiated as in Equations (12-13).

$$\begin{aligned} V(D,G) = {} & \mathbb{E}_{x_1,x_2\sim P_{\text{data}}(x_1,x_2)}[\log D(y|x_1,x_2)] \\ & + \mathbb{E}_{z\sim P_z(z)}[\log(1 - D(G(z,c|x_1,x_2)))] \end{aligned} \tag{12}$$

$$L_I(G,Q) = \mathbb{E}_{z\sim P_z(z),c\sim P_c(c)}[\log Q(c|G(z,c|x_1,x_2))] \tag{13}$$

3.3 Model training

The generator and the discriminator modules are both trained in each epoch but with separate loss functions and optimizers. Three types of losses, namely adversarial loss, information loss, and distance loss, are used to instantiate the minimax game defined in Equations (2, 12-13). Both adversarial loss and distance loss target the interpolation result $\hat{y}$, while the information loss targets the reconstructed latent code $\hat{c}$. The adversarial loss represents the ability of the discriminator to differentiate the interpolation result generated by the generator from the ground truth. The distance loss measures the squared average Euclidean distance between the interpolation result and the ground truth. And finally, the information loss measures the difference between the reconstructed latent code $\hat{c}$ and the original c input. In this way, the total loss $\mathscr{L}$ is a weighted sum of these three types of loss functions, as in Equation (14).

$$\mathscr{L} = L_{\text{adv}} + w_{\text{info}} \cdot L_{\text{info}} + w_{\text{dist}} \cdot L_{\text{dist}} \tag{14}$$

where w represents the weight.

In addition, this study uses the average displacement error (ADE), a commonly used evaluation measure, to evaluate the model performance. ADE, as shown in Equation (15), averages the Euclidean distances between the ground truth and the interpolated locations for all tracking points in the gap.

$$\text{ADE}(y, \hat{y}) = \frac{1}{\tau}\sum_{i=1}^{\tau} \|y^i - \hat{y}^i\| \tag{15}$$

where τ is the length of the gap, $\|\cdot\|$ represents a distance metric and we use the Euclidean distance in this study.

4 EXPERIMENTS

The proposed LSTM-GAN trajectory interpolation model is implemented using PyTorch, a high-performance deep learning library written in Python [19].

4.1 Dataset and data preprocessing

We use a subset of the trajectory dataset collected from 35 adult white storks (*Ciconia ciconia*) over five years (2012-2016) [22]. This dataset is publicly available on Movebank[1] [11, 21], an online platform that helps researchers organize, share, and annotate animal movement data. White storks are long-distance migratory avian animals, which spend their winter times in sub-Saharan Africa before migrating back to Eurasia for breeding [22]. The migration dataset has over a million tracking points. Note that this dataset contains only the trajectories of spring (return) migration, but not their fall (outbound) migration.

Since the tracking devices are solar-powered, they record GPS fixes every 5 min in good solar conditions (95% of the time) or otherwise, every 20 min. At night, the tracking device usually hibernates between 22:00 and 4:00 (for 6 hours). But the exact time might vary according to the solar conditions of that day. Nevertheless, since white storks migrate only during daylight [23, 14], this dataset captures the majority of their movement during migration. To construct continuous trajectories, we connect adjacent tracking points of the same bird to form the trajectories as long as their sampling

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

interval is within the threshold t_{th}. In this case, it is crucial to set a threshold t_{th} that allows ordinary day-to-day time intervals (e.g., 6 hours from 22:00 to 4:00) but meanwhile, identifies abnormal ones (e.g., 20 hours from 21:00 to 17:00) and splits the trajectory there. To achieve this, we first calculate all sampling intervals between adjacent tracking points and then select only large ones (i.e., > 4 h) as they might represent day-to-day time intervals. After that, based on the distribution of these large intervals, we determine that t_{th} = 10 h is an appropriate setting, and it preserves 98.5% of these large intervals. The constructed trajectories are visualized in Figure 3. The white storks start their migration from different wintering sites in Africa but arrive at almost the same location in Europe.

Figure 3: White stork spring migration trajectories

After that, we calculate the speed, heading, and local time of the day for each tracking point to form matrix representations of trajectories. Each trajectory is represented as a $n \times 5$ attribute matrix, where n denotes the number of tracking points and the five attributes chosen in this study are relative movements in x, y directions (denoted as $\Delta x, \Delta y$), speed in x, y directions (denoted as v_x, v_y, containing information on both speed and heading), and time of the day. In addition to the first four commonly used attributes in movement modeling, we also include the time of the day to account for variations in white storks' travel speeds, since they exhibit significantly different speeds depending on the time of day (see Figure 4).

Next, we create trajectory samples by introducing artificial gaps. Following a similar procedure in previous interpolation studies [15, 29, 27], this study uses trajectory samples containing artificial gaps that are manually created to train and evaluate the proposed trajectory interpolation model. In a trajectory sample T_s composed of n tracking points, we place a gap consisting of $n/2$ points in the middle, as in Equation (16). In this study, we create trajectory samples of a variety of lengths to conduct a comparative sensitivity analysis (see Section 4.2.2). Then, following the machine learning tradition, we split the trajectory samples of each length into the training set, validation set, and test set according to a ratio of 0.7 : 0.2 : 0.1.

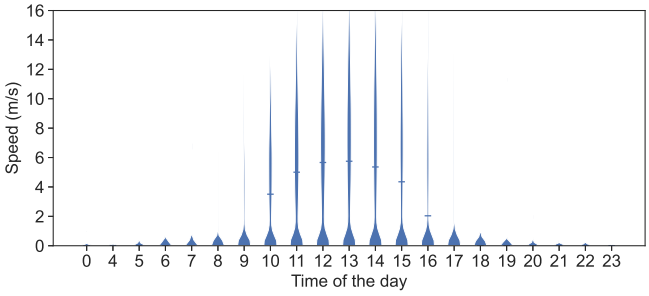

Figure 4: Speed distribution per time of the day with median speeds represented as bars.

$$T_s = \{\underbrace{p^1, \ldots, p^{\frac{n}{4}}}_{x_1:\ \frac{n}{4}\ \text{points}}, \underbrace{p^{\frac{n}{4}+1}, \ldots, p^{\frac{3n}{4}}}_{y:\ \frac{n}{2}\ \text{points}}, \underbrace{p^{\frac{3n}{4}+1}, \ldots, p^{n}}_{x_2:\ \frac{n}{4}\ \text{points}}\} \tag{16}$$

4.2 Sensitivity analysis

4.2.1 Model hyperparameters. Three hyperparameters that may play a significant role are the number of LSTM units, noise dimension, and latent code dimension. Following the common strategy in developing LSTM-based or GAN-based models [18, 8], we experiment with the number of LSTM units in a range from 16 to 256, the noise dimension from 8 to 128, and with the latent code dimension from 1 to 16. As for the weights of loss functions in Equation (14), we use $w_{\text{info}} = 0.5$ as recommended in [2] and we find that setting $w_{\text{dist}} = 1$ gives a promising start in the early training stage as the distance loss helps guide the generator into the right direction.

In the sensitivity analysis of model hyperparameters, we use 10000 trajectory samples, each containing 200 tracking points in total with the middle 100 points missing as a gap (i.e., $|x_1| = |x_2| = 50, |y| = 100$). The models with each setting are trained 100 epochs, and the best model, i.e., the model with the smallest ADE, is selected as representative of that setting. The results are shown in Figure 5, suggesting that the optimal model hyperparameter setting is 128 LSTM units, 32 noise dimensions, and 2 latent code dimensions, where the validation ADE is minimum.

4.2.2 Trajectory sample length. This study applies a comparative sensitivity analysis to evaluate how the length of input trajectory might impact the outcomes. The trajectory sample lengths of 50, 100, 200, 400, 800, and 1600 tracking points are considered. Each input trajectory includes a gap in the middle, with a length of half of the total sample length (see Equation (16) for illustration). The length sensitivity analysis results are presented in Table 1. The overall trend of the relationship between model performance and trajectory length is clear and intuitive. The proposed model has the worst performance on the shortest trajectory samples (i.e., 50 points). This is reasonable because there might not be enough information contained in such short trajectories to accurately interpolate

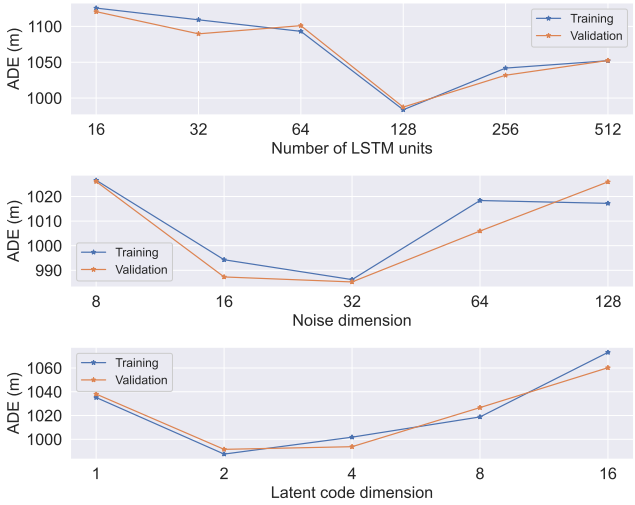

Figure 5: Sensitivity analysis of the number of LSTM units, noise dimension, and latent code dimension

the gap. Then, with the increase in trajectory lengths, it performs better and yields the best result on trajectory samples composed of 200 tracking points. After that, the model performance starts decreasing with a further increase in trajectory lengths. This indicates that the model might be overwhelmed and confused by the information contained in long and complex trajectories. In the following experiments of this study, we continue with trajectory samples with 200 points.

4.3 Model evaluation

4.3.1 Qualitative and quantitative evaluation. After the first 100 epochs of training with distance loss weight $w_{\text{dist}} = 1$, the best model obtained has a validation ADE of 987.02 m for the trajectory length of 200 points. At this early training stage, distance loss provides crucial enlightenment to the generator, guiding it to imitate the ground truth. Hence, we set a large weight for it. After that, our goal is to motivate the generator to generate diverse interpolation results covering a larger probability space when given different latent codes c. In this case, distance loss is restraining the model's ability to deal with uncertainty in movement path choices. Therefore, on the basis of the best model from the first 100 epochs, we train the model for another 100 epochs with $w_{\text{dist}} = 0.5$, and the best model in this second round of training has a validation ADE of 968.03 m. After that, no better model can be obtained even if we continue to this model or further decrease w_{dist} to 0.1, indicating that the model training has converged. Therefore, we complete the model training process and obtain the best model with a validation ADE of 968.03 m. Testing it on the test set, we get a test ADE of 929.67 m, which validates the model performance on trajectory interpolation.

The proposed LSTM-GAN model generates diverse interpolation results with different latent codes c. Therefore, it is crucial to determine the number of applications n_a the proposed model needs in order to cover a reasonably large probability space without wasting computational resources. To analyze that, we experiment with a range of n_a values, and for each trajectory sample, we calculate the average of the minimal ADE among n_a applications with different random noise and latent codes. From the results shown in Figure 6 we can tell that the average minimal ADE drops sharply when the number of applications n_a increases from 1 to 5, and then levels off with further increases of n_a. This indicates that applying the model 5 times with different noise and latent codes can effectively get better approximations of the gap ground truth, that is, the ground truth is more likely to lie within the spatial boundary defined by 5 interpolation results. Thus, in the following experiments, we set $n_a = 5$.

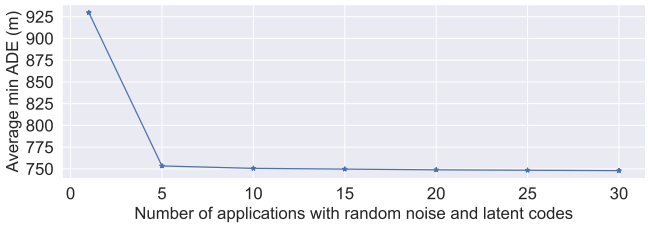

Figure 6: Analysis of the number of applications needed

Six examples of the interpolation results using the model trained on a sample length of 200 points are demonstrated in Figure 7. The observed segments in each trajectory sample are shown in gray, the ground truth during the gap in blue, and the model predictions of the missing points in red. A green circle marks the start of a gap and a yellow triangle marks its end. In each example, the proposed LSTM-GAN model is applied 5 times to create 5 possible interpolation results. Since the trajectory samples are created based on the number of points in the observed segments x_1, x_2 and gap y instead of the distance traveled, it is possible that some of these segments are quite short in terms of the distance traveled. As shown in Figure 7 (a-d), although either or both the observed segments are short, providing minimal useful information, the proposed model manages to generate some reasonable interpolation results for the gap. An extreme case is shown in Figure 7 (d) where both x_1 and x_2 are quite short, but the model still generates some interpolation results that are similar in shape to the ground truth. However, when the tortuosity of the trajectory is large (see Figure 7 (e-f)), the model becomes less accurate. The trajectory sample in Figure 7 (f) has the largest tortuosity, leading to a conspicuous deviation between the gap ground truth and the interpolation results. Overall, the proposed model is able to generate reasonable interpolation results when encountered with trajectories with conspicuously different shapes. However, the interpolation results generated by the proposed model have a simpler geometric shape than that of the ground truth.

4.3.2 Comparative evaluation. To further evaluate the proposed LSTM-GAN model, we compare it against commonly used interpolation methods and present the results in Table 2. The proposed model outperforms linear interpolation and curve-fitting-based interpolation methods when the model is applied only once. When we apply the model $n_a = 5$ times with random noise and latent codes, the model obtains a smaller average minimal ADE, in which case

Table 1: Sensitivity analysis of trajectory sample length

# points in trajectories	Average length (m)	Median length (m)	Training ADE (m)	Validation ADE (m)	Training time (h)
50	38,388	11,214	1133.28	1151.91	0.75
100	79,234	53,163	1131.83	1091.23	1.28
200	**160,995**	**158,816**	**1006.64**	**986.94**	**2.22**
400	325,259	324,725	1014.50	998.68	4.17
800	656,987	660,577	1065.03	1064.73	8.11
1600	1,329,767	1,332,549	1096.10	1107.98	15.83

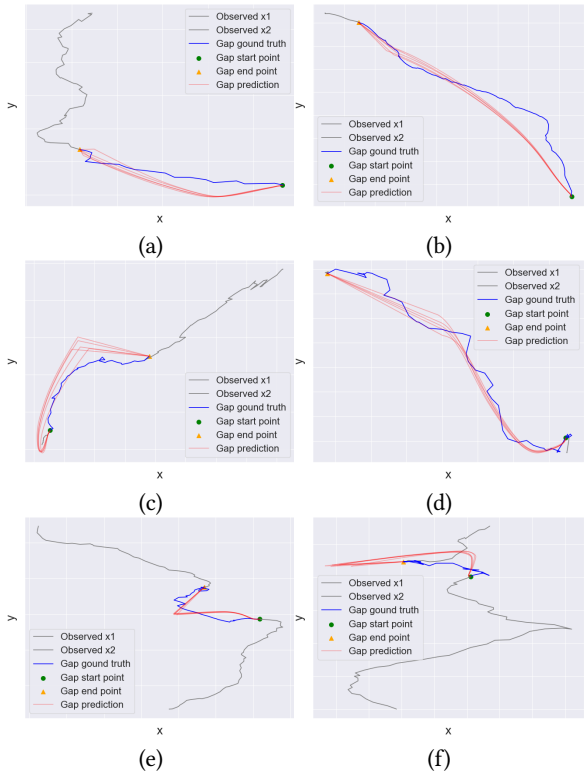

Figure 7: Trajectory interpolation results

we can further approximate the gap ground truth. The proposed model also outperforms the one with traditional GAN architecture, especially when the model is applied multiple times, proving the effectiveness of using an InfoGAN. The results generated by a traditional GAN are highly similar, whereas an InfoGAN learns diverse route choice behaviors using the latent code.

5 DISCUSSION AND FUTURE WORK

The movement patterns are key in a trajectory interpolation task, which can be multifarious. Examples of them include how "exploratory" a moving entity is, corresponding to the tortuosity of the trajectory, or how fast an entity tends to travel at different times of the day, which exhibits both commonalities of a species and characteristics of an individual. These patterns determine how entities choose their movement paths, which is also a decision-making process. Traditional statistical methods are known to struggle when modeling decision-making because the associations between various factors and the outcome are nuanced and entangled, making it difficult to form rigorous representations. In this study, to get around that obstacle, the proposed model learns directly from the data and is able to achieve promising performance given the volume of the trajectory dataset is large enough. In this manner, the movement patterns, as well as the path choice decision-making process, are successfully modeled. The model's ability to deal with uncertainty in movement behaviors comes with adding a latent code c, transforming a traditional GAN into an InfoGAN. The introduction of c enables the model to generate diverse interpolation results given the same pair of observed segments (x_1, x_2). However, in the current phase, the ADE obtained is still large. There is still much room for improvement to generate realistic interpolations. In addition, a major demerit that comes with a deep learning model and cannot be neglected is that the model does so without much interpretability, which is a common demerit of almost all deep learning models. It means that even though the model outputs some trajectory interpolation results, little does the model tell us why the moving entity makes that path choice decision. Therefore, in future work, it is important to study the interpretability of movement models.

Table 2: Comparative evaluation results

Method	ADE (m)	Min ADE (m) when $n_a = 5$
Linear	4525.48	/
Spline (degree:3)	3210.87	/
LSTM + traditional GAN	1124.57	1103.12
LSTM + InfoGAN (Proposed)	**929.67**	**753.35**

In this study, we use the migration trajectories of white storks. During a migration process, a white stork has a clear origin and destination (which are different from each other). An interpolation task in this scenario is in fact studying how a moving entity chooses the movement path from an origin (start of the gap) to a destination (end of the gap). Although the movement path is almost never a straight line, it is somewhat rectilinear. Studying interpolation in this scenario provides guiding implications for developing decision-making models targeted at movement path choices. To make such models more realistic, future extensions may take into account the environmental and social contexts, as they play an important role in impacting the decision-making process [4, 25]. Nonetheless, it

should be noted that there exist movements with other types of patterns. For example, a tiger patrols in its home range, or a marine animal hunts in regular geometric shapes. In these movements, what is important is the exploring behavior during the process (e.g., hunting, patrolling) instead of reaching some destination. Thus, such movements pose a greater challenge for interpolation models or movement path decision-making models in general, since the movement paths in this scenario are less predictable and exhibit a higher level of randomness. Our future work aims to include such movements.

6 CONCLUSION

This study introduces an uncertainty-aware trajectory interpolation model with a generative adversarial network (GAN) architecture using long short-term memory (LSTM) to interpolate trajectory gaps (i.e., spatially and temporally missing tracking data) in wildlife movement data. It uses an InfoGAN to produce samples with variability to deal with uncertainty in movement path choices, which is controlled by a latent code, from the predictive distribution of each individual trajectory. The evaluative experiments suggest promising results indicating the effectiveness of the proposed trajectory interpolation model in real animal tracking datasets. Future work will be devoted to assessing and further enhancing the model for handling tracking data of various sampling rates and lengths, and more complex structures of trajectory data.

ACKNOWLEDGMENTS

Dodge gratefully acknowledges the support from the National Science Foundation through award BCS # 2043202 "CAREER: Modeling Movement and Behavior Responses to Environmental Disruptions".

REFERENCES

[1] Sean C. Ahearn, Somayeh Dodge, Achara Simcharoen, Glenn Xavier, and James L.D. Smith. 2017. A context-sensitive correlated random walk: a new simulation model for movement. *International Journal of Geographical Information Science*, 31, 5, 867–883. DOI: 10.1080/13658816.2016.1224887.

[2] Javad Amirian, Jean-Bernard Hayet, Ac México, and Julien Pettré. 2019. Social Ways: Learning Multi-Modal Distributions of Pedestrian Trajectories with GANs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*.

[3] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, 2172–2180. https://arxiv.org/abs/1606.03657..

[4] Somayeh Dodge et al. 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. DOI: 10.1098/rstb.2013.0195.

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. https://www.deeplearningbook.org/.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, 2672–2680. http://www.github.com/goodfeli/adversarial.

[7] Shaoqing Guo, Junmin Mou, Linying Chen, and Pengfei Chen. 2021. Improved kinematic interpolation for AIS trajectory reconstruction. *Ocean Engineering*, 234, May, 109256. DOI: 10.1016/j.oceaneng.2021.109256.

[8] Agrim Gupta, Justin Johnson, Li Fei-fei Silvio, and Savarese Alexandre. 2018. Social GAN : Socially Acceptable Trajectories with Generative Adversarial Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2255–2264.

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9, 8, 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[10] Jon S. Horne, Edward O. Garton, Stephen M. Krone, and Jesse S. Lewis. 2007. Analyzing animal movements using Brownian bridges. *Ecology*, 88, 9, (Sept. 2007), 2354–2363. DOI: 10.1890/06-0957.1.

[11] Bart Kranstauber, A Cameron, R Weinzerl, T Fountain, S Tilak, M Wikelski, and R Kays. 2011. The Movebank data model for animal tracking. *Environmental Modelling and Software*, 26, 6, 834–835. DOI: 10.1016/j.envsoft.2010.12.005.

[12] Bart Kranstauber, Roland Kays, Scott D Lapoint, Martin Wikelski, and Kamran Safi. 2012. A dynamic Brownian bridge movement model to estimate utilization distributions for heterogeneous animal movement. *Source: Journal of Animal Ecology*, 81, 4, 738–746. DOI: 10.1111/j.l365-2656.2012.01955.x.

[13] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521, 7553, 436–444. DOI: 10.1038/nature14539.

[14] Yossi Leshem and Yoram Yom-Tov. 1996. The use of thermals by soaring migrants. *Ibis*, 138, 4, 667–674. DOI: 10.1111/j.1474-919X.1996.tb04768.x.

[15] Jed A Long. 2016. Kinematic interpolation of movement data Kinematic interpolation of movement data. *International Journal of Geographical Information Science*, 30, 5, 854–868. DOI: 10.1080/13658816.2015.1081909.

[16] Harvey J. Miller. 2005. A measurement theory for time geography. *Geographical Analysis*, 37, 1, 17–45. DOI: 10.1111/j.1538-4632.2005.00575.x.

[17] Ran Nathan, Wayne M Getz, Eloy Revilla, Marcel Holyoak, Ronen Kadmon, David Saltz, and Peter E Smouse. 2008. A movement ecology paradigm for unifying organismal movement research. *Proceedings of the National Academy of Sciences*, 105, 49, 19052–19059.

[18] Seong Hyeon Park, Byeongdo Kim, Chang Mook Kang, Chung Choo Chung, and Jun Won Choi. 2018. Sequence-to-Sequence Prediction of Vehicle Trajectory via LSTM Encoder-Decoder Architecture. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2018-June, Iv, 1672–1678. ISBN: 9781538644522. DOI: 10.1109/IVS.2018.8500658.

[19] Adam Paszke et al. 2019. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, NeurIPS, 8024–8035.

[20] Luca Rossi, Marina Paolanti, Roberto Pierdicca, and Emanuele Frontoni. 2021. Human trajectory prediction and generation using LSTM models and GANs. *Pattern Recognition*, 120, (Dec. 2021). DOI: 10.1016/j.patcog.2021.108136.

[21] Shay Rotics et al. 2018. Data from: Early arrival at breeding grounds: causes, costs and a trade-off with overwintering latitude. *Movebank Data Repository*. DOI: 10.5441/001/1.v8d24552.

[22] Shay Rotics et al. 2018. Early arrival at breeding grounds: Causes, costs and a trade-off with overwintering latitude. *Journal of Animal Ecology*, 87, 6, 1627–1638. DOI: 10.1111/1365-2656.12898.

[23] Shay Rotics et al. 2016. The challenges of the first migration: movement and behaviour of juvenile vs. adult white storks with insights regarding juvenile mortality. *Journal of Animal Ecology*, 85, 4, (July 2016), 938–947. DOI: 10.1111/1365-2656.12525.

[24] J. Marcus Rowcliffe, Chris Carbone, Roland Kays, Bart Kranstauber, and Patrick A. Jansen. 2012. Bias in estimating animal travel distance: The effect of sampling frequency. *Methods in Ecology and Evolution*, 3, 4, 653–662. DOI: 10.1111/j.2041-210X.2012.00197.x.

[25] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. 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. ISBN: 9781728132938. DOI: 10.1109/CVPR.2019.00144.

[26] Georgios Technitis, Walied Othman, Kamran Safi, and Robert Weibel. 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. DOI: 10.1080/13658816.2014.999682.

[27] Yann Tremblay et al. 2006. Interpolation of animal tracking data in a fluid environment. *Journal of Experimental Biology*, 209, 1, 128–140. DOI: 10.1242/jeb.01970.

[28] Zijian Wan, Lianying Li, Huafei Yu, and Min Yang. 2022. A Long Short-Term Memory-Based Approach for Detecting Turns and Generating Road Intersections from Vehicle Trajectories. *Sensors*, 22, 18, (Sept. 2022). DOI: 10.3390/s22186997.

[29] Elizabeth A Wentz, Aimee F Campbell, Robert Houston, Elizabeth A Wentz, Aimee F Campbell, and Robert Houston A. 2003. A comparison of two methods to create tracks of moving objects : linear weighted distance and constrained random walk. 17, 7, 623–645. ISBN: 1365881031000. DOI: 10.1080/13658810310001035492.

[30] Byunggu Yu and Seon Ho Kim. 2006. Interpolating and Using Most Likely Trajectories in Moving-Objects Databases. In *Database and Expert Systems Applications. DEXA 2006. Lecture Notes in Computer Science, vol 4080. Springer, Berlin, Heidelberg*. 718–727. DOI: 10.1007/11827405{_}70.

Received 26 September 2023; revised 13 October 2023; accepted 16 October 2023