

ENCODING AGENT TRAJECTORIES AS REPRESENTATIONS WITH SEQUENCE TRANSFORMERS

Athanasios Tsiligkaridis, Zhongheng Li, & Elizabeth Hou

Systems & Technology Research (STR)

Woburn, MA 01801, USA

{athanasios.tsiligkaridis, zhongheng.li, elizabeth.hou}@str.us

ABSTRACT

Spatiotemporal data faces many analogous challenges to natural language text including the ordering of locations (words) in a sequence, long range dependencies between locations, and locations having multiple meanings. In this work, we propose a novel model for representing high dimensional spatiotemporal trajectories as sequences of discrete locations and encoding them with a Transformer-based neural network architecture. Similar to language models, our Sequence Transformer for Agent Representation Encodings (STARE) model learns these encodings through the supervisory signal of various tasks, e.g. classification of agent trajectories. We present experimental results on various synthetic and real trajectory datasets and show that our proposed STARE model can correctly learn labels along with meaningful encodings.

1 INTRODUCTION

In the modern world, geospatial mobility data has become increasingly available with the proliferation of mobile devices and other positioning and sensor technologies (Zheng et al., 2011; PFLOW, 2019). These technologies have also allowed wildlife researchers to monitor and collect data on animal movements and their ecosystems (BirdLife, 2023; Movebank, 2023). This increasing availability of location datasets has been leveraged by researchers allowing them to build models that further their understanding of mobility patterns (Jiang & Luo, 2022; Bartlam-Brooks et al., 2013), trafficking, and infectious disease (Hu et al., 2021; Ratanakorn et al., 2018) (in both humans and animals). Inspired by the successes of neural networks in other fields (e.g. vision, language, speech), researchers have leveraged their flexible architecture and applied them to trajectory data to model human and animal mobility patterns (Luca et al., 2021; Wijeyakulasuriya et al., 2020).

The sequential nature of trajectory data shares many similar properties with the sequential nature of natural language. Tokenizing sentences or text segments in Natural Language Processing (NLP) is essentially a mapping from a complex “quasi-infinite” space to a sequence of discrete elements in a finite vocabulary. In a similar fashion to rule-based or algorithmic tokenizers (Honnibal & Montani, 2017; Sennrich et al., 2015), the “quasi-continuous” high frequency GPS coordinates in trajectory data can also be tokenized into a lower dimensional sequence with a finite vocabulary. Also, just as tokens in language settings have semantic meaning (e.g., [*fir*] refers to a coniferous tree), tokens in mobility data can also have inherent meaning about the location they represent (e.g., (45.832119, 6.865575) is in the French alps). The order of the tokens in both domains also contains key information as language tokens can modify each other and location tokens can have different implicit meanings depending on their order (e.g., visiting a supermarket before work makes it a breakfast location whereas after work makes it a grocery shop). We can further extend our analogy between NLP and trajectory data to the “document” level where just as sentences in the same document will share similar properties such as word choice, grammar, and style, trajectories collected from the same agent or user will also share similar orderings of locations or routes. Thus, we leverage these numerous similarities by proposing the Sequence Transformer for Agent Representation Encodings (STARE) model, a neural network with an encoder-based transformer architecture similar to those used in BERT-like language models (Devlin et al., 2018; Vaswani et al., 2017).

Other works have also observed the connection between the sequential nature of trajectory data and language or other sequential domains. Earlier works focused on more simple architectures, such as RNN in De Brébisson et al. (2015) and LSTM in Xu et al. (2018), to learn to predict trajectories and final locations. While later works also leveraged the power of transformer style architectures, Tsiligkaridis et al. (2020); Giuliani et al. (2021); Abideen et al. (2021) they focused on forecasting trajectories for next destination prediction. Additionally, these transformer-based architectures also added significant amounts of side information, in the form of a contextual block (Tsiligkaridis et al., 2022) or multiple feature extractors that incorporate a points of interest ontology and a social network (Xue et al., 2021). Instead of just having a similar architecture, Li et al. (2022) and Xue et al. (2022) extended pre-trained large language models to take in pseudo-sentences that represent locations in natural language, effectively performing an NLP task.

In contrast, we propose an approach that does not require any additional information aside from timestamped spatiotemporal data. With this, our STARE model has the flexibility to be used in a variety of settings where only location and time data is present, such as with various animal trajectory datasets and in environments without contextual and/or foundational information. Our model simply takes in *only* spatiotemporal data and ultimately learns important encodings that can be used to both solve various downstream tasks (e.g., classification, destination prediction, clustering) and learn recurring behaviors and Patterns of Life (PoL) of various agents of interest.

In this work, we make the following contributions. First, we present a data discretization technique for reducing the dimensionality of long and rich agent PoL data. Second, we propose a novel transformer-based architecture for obtaining informative data embeddings which can be used to learn relationships between agents. Finally, we present experiments on both simulated and real trajectory datasets to demonstrate our model’s ability to learn informative encodings of sequential data via a classification task.

2 STARE MODEL

In this section, we describe our STARE model, which compresses raw trajectory data into novel tokenized sequences for input into a Transformer Encoder Stack (TES). Our architecture is similar to that of Tsiligkaridis et al. (2022), but we specifically focus on the minimal data setting where we solely have sequence information as data, i.e. we do not use a contextual block in our input as we do not incorporate any non-sequential information in our model. The aim of STARE is to learn encodings of the trajectory data that have semantic meaning and, in turn, can be used to predict labels of interest with a Multi-Layer Perceptron (MLP). Figure 1 displays our architecture.

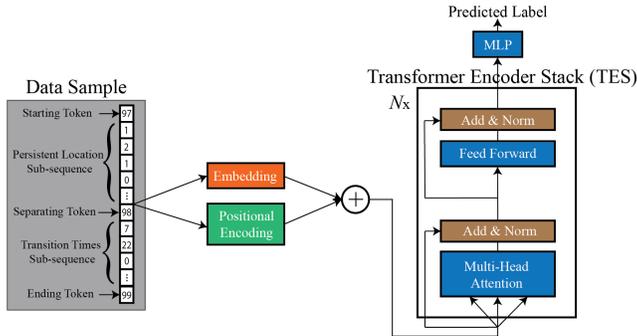


Figure 1: Our proposed STARE Transformer-based architecture which uses a TES to create meaningful embeddings of input data and an MLP to make class membership predictions.

We begin with a dataset \mathcal{X} containing N observations of a tuple containing the latitude, longitude, and timestamp of each agent for A total agents. To form multiple samples per agent and to reduce the size of our model architecture, we make independence assumptions on the temporal component by assuming that timepoints within a time window are dependent with each other, but independent of those outside the window. Explicitly, we partition X_a , the trajectory of agent a , into set of M sub-trajectories, $\{X_1, X_2, \dots, X_M\}$ where each sub-trajectory is the length of some time window (e.g. 6 hours, a day, or a week). The choice of the time window length is dataset dependent, but

uses the implicit assumption that repetitive behavior is expected and independent of each other. For example, humans tend to have cycles of the same behavior over days (e.g., people live in one place, wake up, go to work, do some activities throughout the day, and then return home to end their day) with some seasonality and anomalies. These independence assumptions are very typical in the sequential domain and parallel the breaking of text into multiple samples between sentences in NLP.

For a given agent $a \in [1, \dots, A]$ and a time window $m \in [1, \dots, M]$, we define a trajectory $T_{a,m}$ as:

Definition 1 (Trajectory $T_{a,m}$) For a time window m , agent a has a raw trajectory $T_{a,m}$ that is defined as a sequence of $L_{a,m}$ time-stamped locations: $T_{a,m} = [p_1, \dots, p_{L_{a,m}}]$, where each point $p_i = [\text{lat}_i, \text{lon}_i, t_i]$, is a tuple of latitude, longitude, and time, that identifies the geographic location of that point p_i at time t_i .

The number of data points in $T_{a,m}$ can be very large and varying between different agents and time windows due to potential uneven measurement sampling rates and durations. Since locations are defined by their quasi-continuous latitude and longitude values, there is an infinitesimally large number of unique locations. To combat this issue, we discretize the data inputs by only retaining information about an agent’s Persistent Locations (PL) and the times they transition between PLs.

Specifically, we define a PL as a stationary point in a trajectory that is mapped to a discrete value in an alphabet, where we define this alphabet to be S2 cells (a hierarchy of indexed spatial cells that represent geographical areas over the world) at a certain zoom level (S2 Developers, 2024). We can also discretize the transition times between PLs by defining another alphabet that is multiples of some time measure, (e.g. minutes, hours) and rounding the transition times to the nearest multiple. Let $s : [\Phi, \Lambda] \rightarrow \mathbb{S}^2$ be a mapping from latitude, longitude space to our S2 cell alphabet and $tt : \mathbb{R}^+ \rightarrow \mathbb{T}$ be a mapping from the positive real space of transition times to our discrete alphabet of rounded times. Thus, the concatenated sub-sequences of visited PLs and transition times form a data sample for agent a on time window m that is a discretized and compressed version of $T_{a,m}$,

$$x_{a,m} = \left[[BOS], s(T_{a,m}), [SEP], tt(T_{a,m}), [EOS] \right] \quad (1)$$

where the $[BOS]$, $[EOS]$, $[SEP]$ tokens, borrowed from NLP, represent beginning, end, and separating tokens, respectively. We show an example of this process in Appendix A.1. Evidently, our discretization is a means of tokenizing our input to be passed into our Transformer-based model.

3 SIMULATION AND WILDLIFE ANIMAL MOVEMENT EXPERIMENTS

We simulate trajectories with a generative model that approximately aligns with our modelling assumptions for the STARE model. Specifically, our generative model is parameterized by a series of PLs associated with each agent. We also make the following assumptions in order to have realistic behaviors for the *human* agents: They often: 1) have reoccurring behaviors in the same locations (e.g. go to the same house every day), 2) live and behave similarly to others (i.e. there is some sharing of PLs between humans), and 3) travel along roads between their PLs.

We randomly assign PL locations to each agent using foundational data from ORNL (2024); Thakur et al. (2015); OSM contributors (2017) where the assignments encourage similarity among groups of agents (e.g. their houses are within some radius of each other or they share the same office location). We then connect the PLs using an Open Street Maps (OSM) road network from the OSMnx python package (Boeing, 2017) in order to have a high frequency dataset. For our experiments below, we simulated data trajectories with 10, 20, and 30 subpopulations with 37, 348, and 1288 agents total, respectively, over a 28 day time period.

We showcase the performance of our STARE model with a 4 attention head, 2 encoder stack architecture and use an 85%/15% train/test split. We tokenize each agent’s trajectories into a 1 day time window, and map the stationary points to an alphabet based on Zoom 16 S2 cells and the transition times to an alphabet of 30 minute increments. Since we know which agents belong to which subpopulations, we can train our STARE model to learn either subpopulation membership or agent behaviors using the corresponding labels.

In Table 1, we show the Correct Classification Rates (CCR), i.e. accuracy, for each training approach on the 3 simulated datasets. Additionally, we compare our method against single-stack LSTM and BiLSTM baselines with embedding and hidden dimensions of sizes 128 and 64, respectively; we see

that our STARE model outperforms these baselines. When training on subpopulation labels, STARE has perfect CCRs, but when training on agent labels, there are some misclassifications due to the similar PoLs between agents in the same subpopulation. Figure 2 represents a matrix of averaged predicted probability scores for the (S) synthetic dataset where each row is an agent class. We see that many agents are correctly identified (only non-zero value is in the diagonal), but there are others that have multiple blocks of non-zero values. Figure 3 visualizes the trajectories of the agents in the first two blocks where we can see the reason for the misclassification: the PoLs between these agents are extremely similar. Thus, we can use these misclassifications to learn relationships between agents. In many realistic scenarios, subpopulation information will not be present, so we can use agent labels to learn informative encodings from the output of the TES and leverage them to learn relationships between agents in an attempt to form subpopulations.

To assess the effectiveness of extracting meaningful encodings with STARE in real scenarios, we applied it to a raw trajectory dataset of ravens (Jain et al., 2022). We only used trajectory data from 49 ravens between March and July for both 2018 and 2019 because we observed a higher volume of data for these intervals (Figure A8 displays the distribution of data in these months). We tokenize each agent’s trajectories into a 1 day time window and train our STARE model using the same model architecture and hyperparameters used in the synthetic data experiments.

We train our STARE model, along with baseline single-stack LSTM/BiLSTM models, using agent labels with the same architecture and train/test split percentages used in the simulated data and obtain the CCRs shown in Table 2. For our model, we then apply Principal Component Analysis (PCA) to our STARE encodings to reduce the dimension of our embeddings and we also visualize the first three principal components in Figure 4, where we color code samples based on agent labels. We see separation between these color-coded point clouds, and each one represents samples of a raven. As an example, we focus on the light blue mass that contains samples of a raven labeled as “Lisa”, which has a wide and distinct PoL, as shown in Figure 5. Given these embeddings, we apply an MLP to get probability scores, we apply Spectral Clustering (SC) on these probabilities, and obtain the result in Figure A9a over the 49 ravens. Here, we identified 4 distinct clusters; in Cluster 2, we see two ravens that exhibit unique movement patterns. Lisa is a member of this cluster and has a distinct PoL, which the SC identified. All ravens are visualized in Figure A9b, and clear distinctions between the 4 clusters can be seen.

4 CONCLUSIONS AND FUTURE WORK

In this work, we proposed a spatiotemporal transformer-based model for learning both labels and meaningful data encodings. Through experiments, we demonstrated our model’s ability to correctly learn labels and informative encodings of sequential data. As future work, we plan to investigate our model’s ability to scale up to very large volumes of data and also train our model with a masked language modeling scheme to further encourage it to learn latent information embedded within trajectory data.

5 ACKNOWLEDGEMENTS

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DOI/IBC) contract number 140D0423C0046. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

Dataset	STARE		LSTM		BiLSTM	
	Subpop	Agent	Subpop	Agent	Subpop	Agent
(S): 10 Subpops, 37 Agents	100%	84.0%	100%	82.6%	100%	80.6%
(M): 20 Subpops, 348 Agents	100%	79.4%	100%	77.6%	100%	79.4%
(L): 30 Subpops, 1288 Agents	100%	74.2%	99.7%	73.2%	99.9%	74.1%

Table 1: CCR (accuracy) performance table of our STARE model, along with baseline single-stack LSTM and BiLSTM architectures, on various simulated datasets and labeling schemes. The *subpop* and *agent* columns represent training on subpopulation and agent labels, respectively. We see that our STARE model slightly outperforms the LSTM baselines in terms of the CCR accuracy metric.

Dataset	STARE	LSTM	BiLSTM
Ravens [Jain et al. (2022)]	39.8%	3.5%	3.9%

Table 2: Performance table of our STARE model, along with baseline single-stack LSTM and BiLSTM architectures, on the *Ravens* dataset where we train with agent labels as subpopulation labels do not exist here.

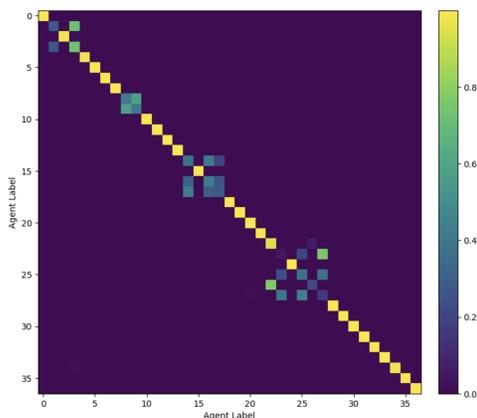


Figure 2: Matrix of avg. predicted probability scores for (S) where rows are the 37 agents.



Figure 3: Visualization of the similar PoIs between misclassified agents. The agents corresponding to the 1st / 3rd rows are in red / yellow and those for the 8th / 9th rows are in blue / white.

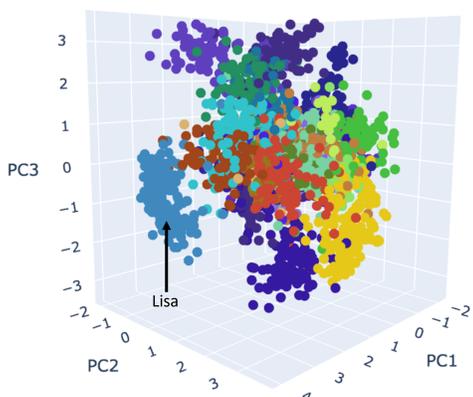


Figure 4: 3D PCA of encodings (colored by agent)

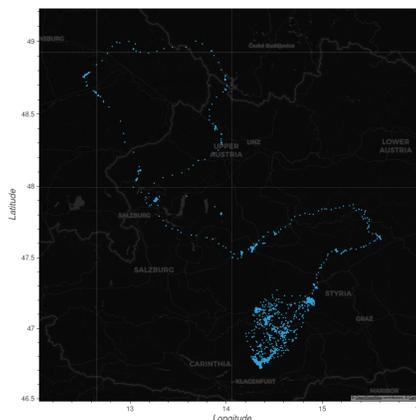


Figure 5: Lisa’s raw trajectories from Cluster 2.

REFERENCES

- Zain Ul Abideen, Heli Sun, Zhou Yang, Rana Zeeshan Ahmad, Adnan Iftekhhar, and Amir Ali. Deep wide spatial-temporal based transformer networks modeling for the next destination according to the taxi driver behavior prediction. *Applied Sciences*, 11(1), 2021. doi: 10.3390/app11010017. URL <https://www.mdpi.com/2076-3417/11/1/17>.
- Hattie L. A. Bartlam-Brooks, Pieter S. A. Beck, Gil Bohrer, and Stephen Harris. In search of greener pastures: Using satellite images to predict the effects of environmental change on zebra migration. *Journal of Geophysical Research: Biogeosciences*, 118(4):1427–1437, 2013. doi: <https://doi.org/10.1002/jgrg.20096>. URL <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/jgrg.20096>.
- BirdLife. Seabird tracking database, 2023. <https://www.seabirdtracking.org/>.
- Geoff Boeing. OSMnx: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks. *Computers, Environment and Urban Systems*, 65:126–139, 2017.
- Alexandre De Brébisson, Étienne Simon, Alex Auvolat, Pascal Vincent, and Yoshua Bengio. Artificial neural networks applied to taxi destination prediction. In *Proceedings of the 2015th International Conference on ECML PKDD Discovery Challenge - Volume 1526, ECMLPKDDDC’15*, pp. 40–51, Aachen, DEU, 2015. CEUR-WS.org.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. Transformer networks for trajectory forecasting. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 10335–10342, 2021. doi: 10.1109/ICPR48806.2021.9412190.
- Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.
- Tao Hu, Siqin Wang, Bing She, Mengxi Zhang, Xiao Huang, Yunhe Cui, Jacob Khuri, Yaxin Hu, Xiaokang Fu, Xiaoyue Wang, et al. Human mobility data in the covid-19 pandemic: characteristics, applications, and challenges. *International Journal of Digital Earth*, 14(9):1126–1147, 2021.
- Varalika Jain, Thomas Bugnyar, Susan J Cunningham, Mario Gallego-Abenza, Matthias-Claudio Loretto, and Petra Sumasgutner. The spatial and temporal exploitation of anthropogenic food sources by common ravens (*corvus corax*) in the alps. *Movement Ecology*, 10(1):1–15, 2022.
- Weiwei Jiang and Jiayun Luo. Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications*, 207:117921, 2022. ISSN 0957-4174. doi: <https://doi.org/10.1016/j.eswa.2022.117921>. URL <https://www.sciencedirect.com/science/article/pii/S0957417422011654>.
- Zekun Li, Jina Kim, Yao-Yi Chiang, and Muhao Chen. Spabert: A pretrained language model from geographic data for geo-entity representation. *arXiv preprint arXiv:2210.12213*, 2022.
- Massimiliano Luca, Gianni Barlacchi, Bruno Lepri, and Luca Pappalardo. A survey on deep learning for human mobility. *ACM Comput. Surv.*, 55(1), nov 2021. ISSN 0360-0300. doi: 10.1145/3485125. URL <https://doi.org/10.1145/3485125>.
- Movebank. Movebank data repository, 2023. <https://datarepository.movebank.org/home>.
- ORNL. Oak Ridge National Laboratory (ORNL); Federal Emergency Management Agency (FEMA) Geospatial Response Office USA Structures, 2024. <https://gis-fema.hub.arcgis.com/pages/usa-structures>.
- OSM contributors. Planet dump retrieved from <https://planet.osm.org>. <https://www.openstreetmap.org>, 2017.

- People Flow Project PFLOW. People flow data, 2019. <https://pflow.csis.u-tokyo.ac.jp/home/>.
- Parntep Ratanakorn, Sarin Suwanpakdee, Witthawat Wiriyarat, Krairat Eiamampai, Kridsada Chai-choune, Anuwat Wiratsudakul, Ladawan Sariya, and Pilaipan Puthavathana. Satellite telemetry tracks flyways of asian openbill storks in relation to h5n1 avian influenza spread and ecological change. *BMC veterinary research*, 14:1–11, 2018.
- S2 Developers. S2 Geometry Library, 2024. <http://s2geometry.io/>.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.
- Gautam Malviya Thakur, Budhu Bhaduri, Jesse Piburn, Kelly Sims, Robert Stewart, and Marie Urban. Planetsense: A real-time streaming and spatio-temporal analytics platform for gathering geo-spatial intelligence from open source data. 11 2015. doi: 10.1145/2820783.2820882. URL <https://www.osti.gov/biblio/1240540>.
- Athanasios Tsiligkaridis, Jing Zhang, Hiroshi Taguchi, and Daniel Nikovski. Personalized destination prediction using transformers in a contextless data setting. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2020. doi: 10.1109/IJCNN48605.2020.9207514.
- Athanasios Tsiligkaridis, Jing Zhang, Ioannis Ch. Paschalidis, Hiroshi Taguchi, Satoko Sakajo, and Daniel Nikovski. Context-aware destination and time-to-destination prediction using machine learning. In *2022 IEEE International Smart Cities Conference (ISC2)*, pp. 1–7, 2022. doi: 10.1109/ISC255366.2022.9922593.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- Dhanushi A Wijeyakulasuriya, Elizabeth W Eisenhauer, Benjamin A Shaby, and Ephraim M Hanks. Machine learning for modeling animal movement. *Plos one*, 15(7):e0235750, 2020.
- Yanyu Xu, Zhixin Piao, and Shenghua Gao. Encoding crowd interaction with deep neural network for pedestrian trajectory prediction. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5275–5284, 2018. doi: 10.1109/CVPR.2018.00553.
- Hao Xue, Flora Dilys Salim, Yongli Ren, and Nuria Oliver. Mobtcast: Leveraging auxiliary trajectory forecasting for human mobility prediction. In *Neural Information Processing Systems*, 2021. URL <https://api.semanticscholar.org/CorpusID:238259192>.
- Hao Xue, Bhanu Prakash Voutharoja, and Flora D Salim. Leveraging language foundation models for human mobility forecasting. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, pp. 1–9, 2022.
- Yu Zheng, Hao Fu, Xing Xie, Wei-Ying Ma, and Quannan Li. Geolife gps trajectory dataset - user guide, July 2011. <https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide/>.

A APPENDIX

A.1 DATA PREPROCESSING:

To obtain PLs, we use the location and time data at hand to create a speed signal for agent a in time window m . With this, we want to locate all of the stationary points where agent a does not move. To do so, we apply a speed threshold $\tau_{\text{speed}} = 5\text{mph}$ and extract all of the coordinates where the speeds are below τ_{speed} . Figure A6 displays the output of this approach for a synthetic dataset where PLs are shown in red and the S2 cells they reside within are shown in yellow.

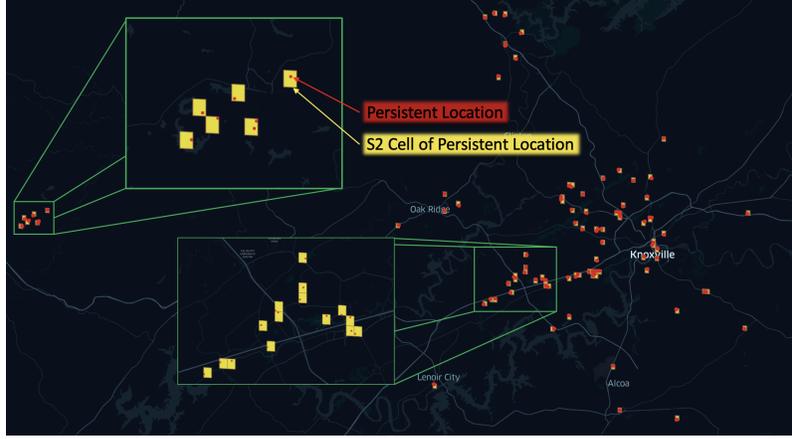


Figure A6: Visualization of all extracted PLs from the 10 *Subpopulations*, 37 *Agents* synthetic dataset along with the S2 cells that each PL resides in.

As a guiding example, Figure A7 displays an agent’s entire PoL along with their PL information. The data in blue represents the PoL defined by their GPS positions, the red represents the PLs, and the yellow rectangles represent S2 cells.

To form a data point for agent a in time window m , we must create visited PL and transition time sub-sequences. We begin with $T_{a,m}$, which is the raw trajectory of agent a in time window m . Let us say that in time window m , agent a starts at PL_1 , goes to PL_2 , then to PL_5 , then to PL_3 , and finally ends the day back at PL_1 . We define the PLs using indices from 1 to the maximum amount of PLs, n_{PL} . The location sequence for the day becomes: $s(T_{a,m}) = [1, 2, 5, 3, 1]$. With this, we then look at the transition times between the visited locations and form a sequence of these times as: $tt(T_{a,m}) = [t_{PL_1 \rightarrow PL_2}, t_{PL_2 \rightarrow PL_5}, t_{PL_5 \rightarrow PL_3}, t_{PL_3 \rightarrow PL_1}]$. Realistically, this sequence could be: $tt(T_{a,m}) = [9\text{AM}, 5\text{PM}, 7\text{PM}, 8\text{PM}]$. Next, we adjust $tt(T_{a,m})$ by discretizing time into 30 minute blocks, assigning integers to each block, and offsetting these indices based on the total amount of PLs in the data (n_{PL}), so as to avoid token overlap in the input between positions and times. With this, our time sequence can become: $tt(T_{a,m}) = [48, 64, 68, 70]$, where each element represents a real time block. Finally, we zero pad $s(T_{a,m})$ and $tt(T_{a,m})$ to the maximum observed lengths in our data (obtained by searching over all created sequences over all agents and days), incorporate start, $[BOS]$, separating, $[SEP]$, and end, $[EOS]$, tokens, and concatenate them all as:

$$x_{a,m} = \left[[BOS], s(T_{a,m}), [SEP], tt(T_{a,m}), [EOS] \right].$$

Realistically, the entire data sequence can be:

$$x_{a,m} = [97, 1, 2, 5, 3, 1, \dots, 0, 98, 48, 64, 68, 70, \dots, 0, 99],$$

where 97, 98, and 99 represent $[BOS]$, $[SEP]$, and $[EOS]$ tokens, respectively.

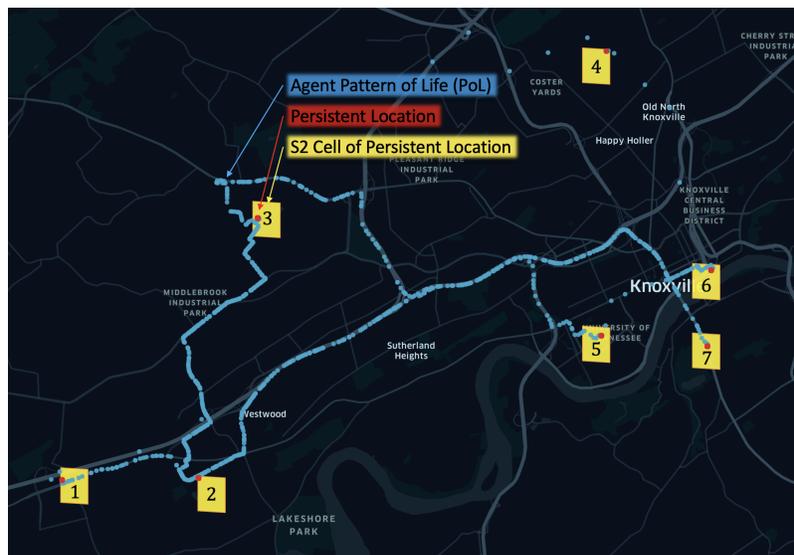


Figure A7: Visualization of an agent's PoL (from the 10 *Subpopulations*, 37 *Agents* synthetic dataset) over a 28 day period along with its derived PLs and the S2 cells that each PL resides in.

A.2 ADDITIONAL FIGURES FOR THE RAVEN DATASET:

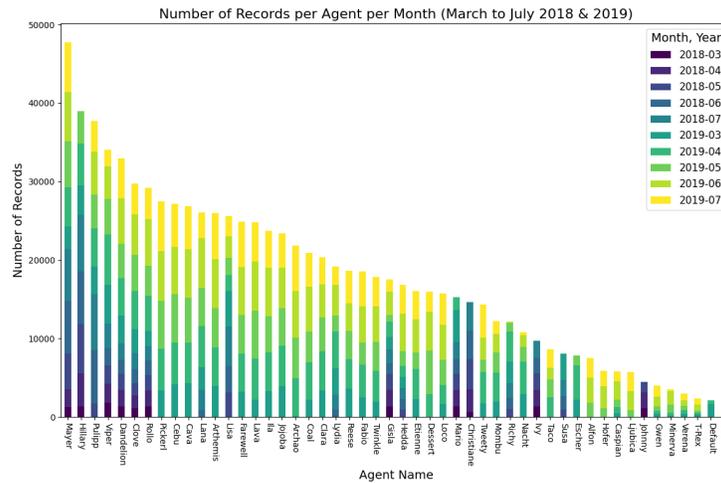
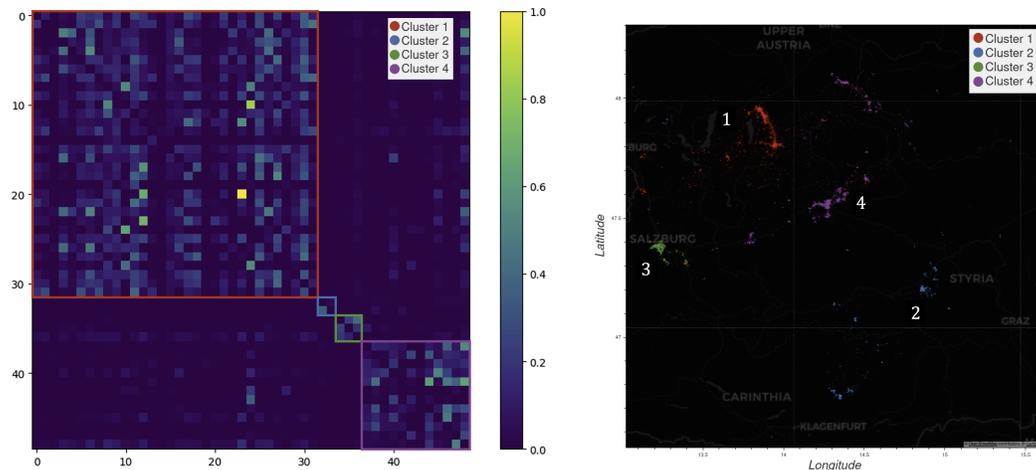


Figure A8: Distribution of data records among individual GPS-transmitter-equipped ravens during the months of March to July for both 2018 and 2019, with at least 100,000 raw trajectory records per month. This period corresponds to the annual temperature rise in the Austrian Alps, from an average of 36.8°F in March to 62.5°F in July, which may formulate the habitual conditions of migration. Even though this data volume graph is skewed, we still observe that ravens can have large amounts of daily visited PLs.



(a) 4 clustered identified with Spectral Clustering were highlighted with color labeled bounding boxes (b) Datashaded Trajectories of all 49 ravens colored by cluster for all 4 clusters as numbered

Figure A9: The averaged predicted probability score matrix is clustered with Spectral Clustering in (a) into 4 clusters (where each cluster is highlighted with a color-labeled bounding box) and each raven’s trajectories are colored by these cluster memberships in (b) where the intensity is dependent on frequency of visits to the location.