Trajectory Prediction Learning Using Deep Generative Models

MSc. Thesis of Jing Li
Department of Electrical Engineering and Computer Science

Dec 19, 2023

YORK U

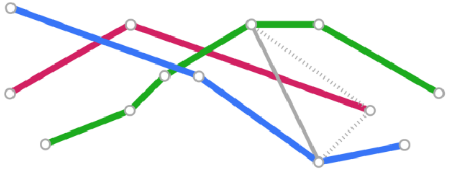
Introduction

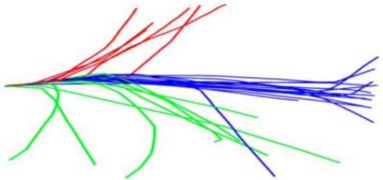
Trajectory/Mobility Data

Trajectory: A Sequence of (Spatiotemporal) Points

Vast Amounts of Trajectory/Mobility Data

Trajectory-related Problems





trajectory similarity

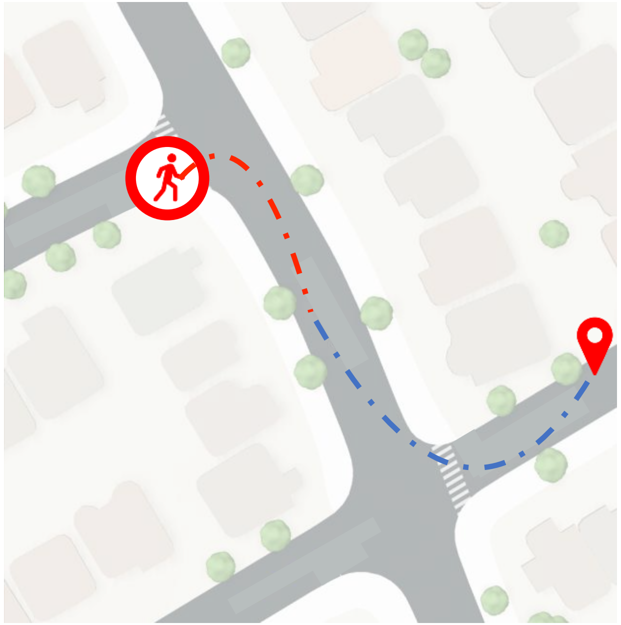
trajectory clustering

trajectory imputation

pedestrian crowd behavior

...

Problem of Interest: Trajectory Prediction



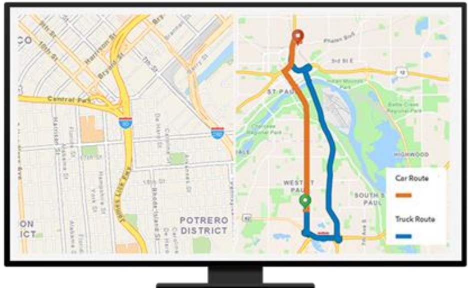
Predict future trajectory

Plethora of Applications

Ride-sharing services

Next POI recommendation

Autonomous vehicles



Traffic flow optimization

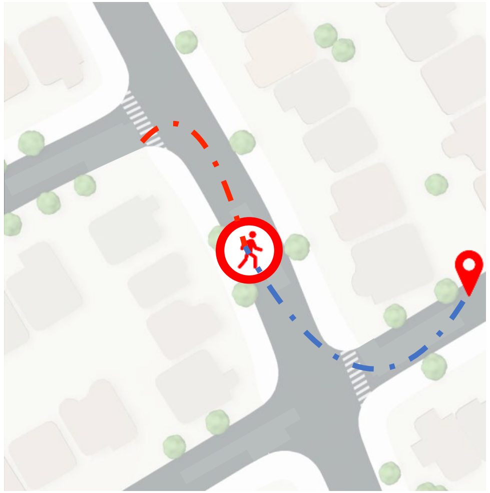
YORK U

Problem Statement

Trajectory Prediction

Let

- an observation area M
- an observation period [0, W]
- a set of objects N and their history trajectories S'



Input: Given

- a moving object i in N
- a partial trajectory $T = \langle p_{i1}, p_{i2}, \ldots, p_{il} \rangle$
- a prediction horizon $k > 0$

Output: We want to

predict the next k spatiotemporal points $\langle p_{il+1}, p_{il+2}, \ldots, p_{il+k} \rangle$ of the partial trajectory T

Overview

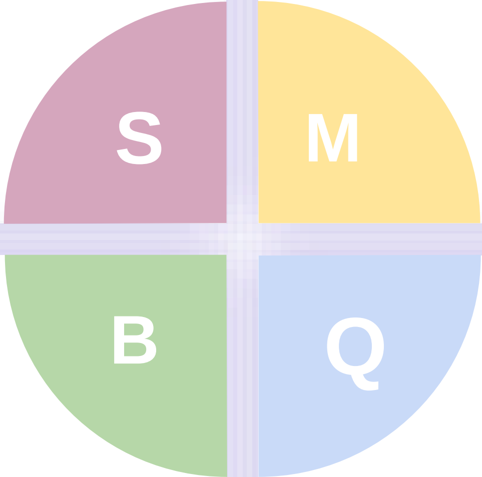
- Higher-order Mobility Flow Data
- (Revisit) Problem Statement
- Existing Works
- Methodology
- Evaluation
- Conclusions

Higher-order Mobility Flow Data

Challenges of Working with Trajectory Data

Data Sparsity

Limited data



Model Incompatibility

Not compatible with well-known machine learning models

80% of the data is generated by 20% of the users

Imbalanced Data

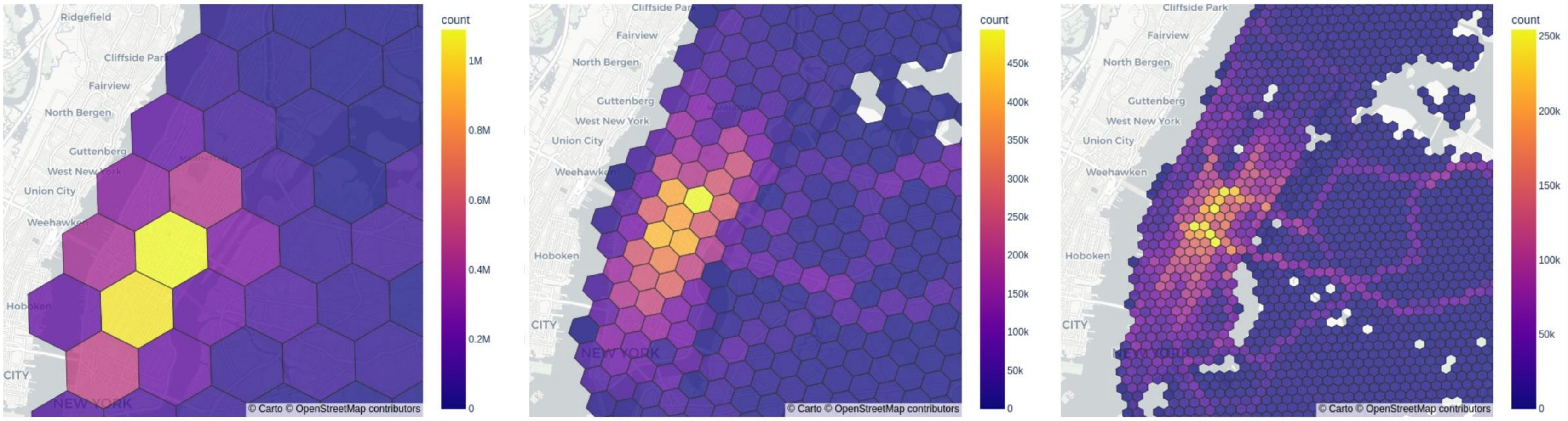
Low accuracy and completeness

Data Quality

YORK U

Map Tessellation

Low resolution → High resolution

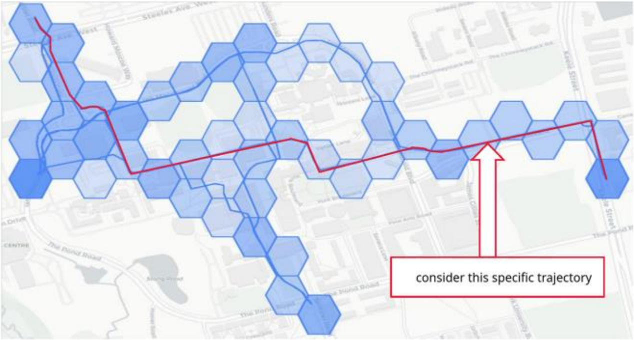


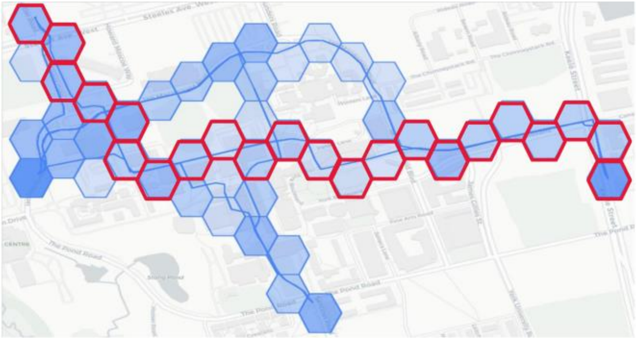
Why hexagons?

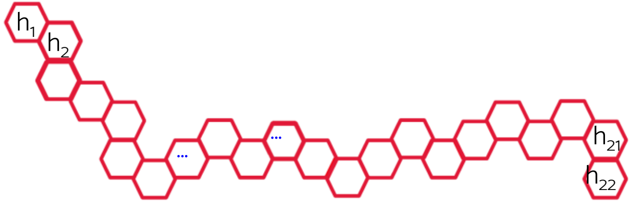
- More circular that fully tessellates the space
- Same distance to all adjacent neighbours

YORK U

Trajectories: Sequences of Hexagons







Trajectory: $h_1, h_2, h_3 \ldots h_{20}, h_{21}, h_{22}$

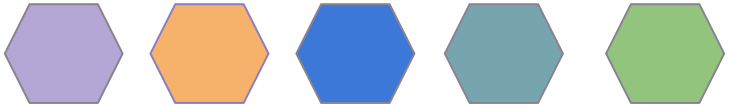
YORK U

Treat Trajectories as Language Statement

Treat Trajectories as Language Statements

Hexagons represent ‘tokens’ & trajectories represent ‘sentences’

Trajectory:

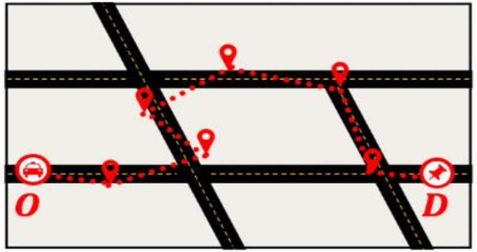


Sentence: I like to learn English

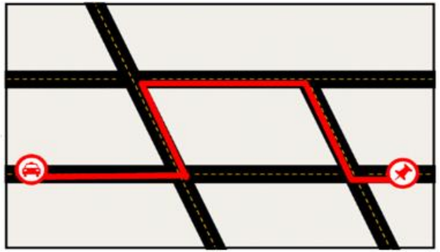
Advantages:

- Reduced data sparsity
- More compatible with well-known ML models (e.g., sequence models, LLMs)

Point2Hex: Overview of the Pipeline

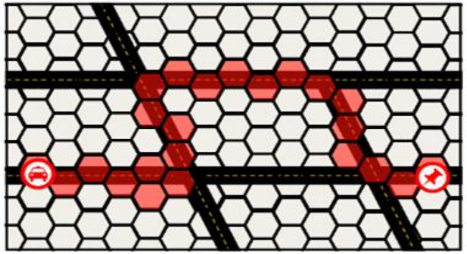


GPS Traces or POI Check-Ins
(**input**)

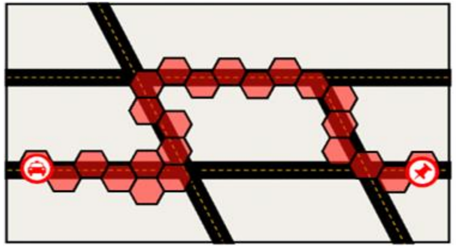


Linestring of Trajectories
(**Map-matching**)

Map Tessellation with Trajectories
(**Hexagon-shaped cells**)



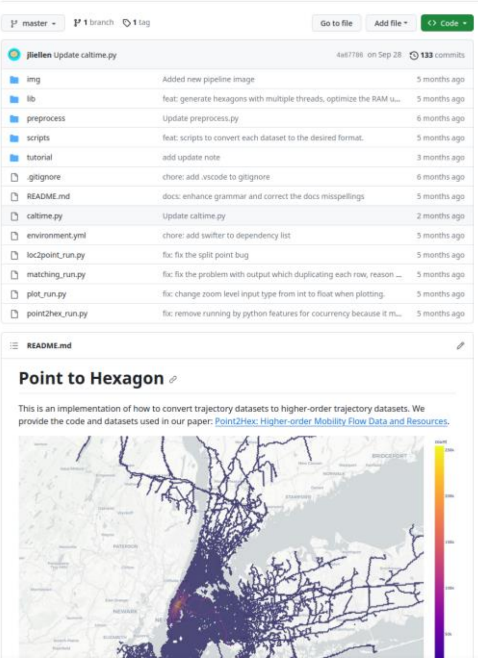
Intersection of Linestrings and Polygons
(**Computational Geometry**)



Higher-order Mobility Flow
(**Output**)

Code of Point2Hex (Data Generator)

The code to generate your HO dataset from raw GPS points



Check it on
[GitHub]

YORK U

Datasets: Higher-order Mobility Flow

Dataset	Trajectories	Time Period	Resolutions
HO-T-Drive	65,117	02/02/08 - 02/08/08	{6,...,10}
HO-Porto	1,668,859	07/01/13 - 06/30/14	{6,...,10}
HO-Rome	5,873	02/01/14 - 03/02/14	{6,...,10}
HO-GeoLife	2,100	04/01/07 - 10/31/11	{6,...,10}
HO-FourSquare-NYC	49,983	04/12/12 - 02/16/13	{6,...,10}
HO-FourSquare-TKY	117,593	04/12/12 - 02/16/13	{6,...,10}
HO-NYC-Taxi	2,062,554	01/01/16 - 06/30/16	{6,...,10}

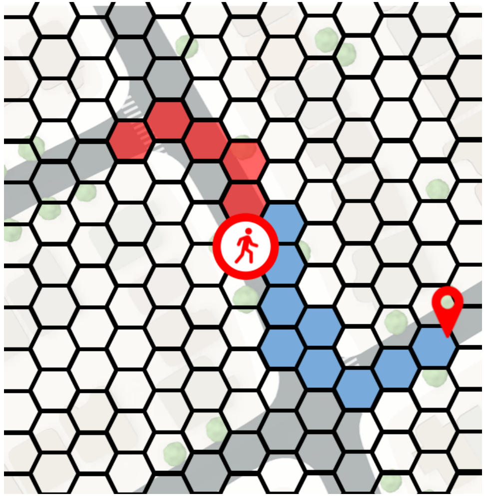
Download from [Zenodo]

(Revisited) Problem Statement

Trajectory Prediction (Revisited)

Let

- an observation area *M*
- an observation period [0, W]
- a set of objects *N* and their history trajectories *S*



Input: Given

- a moving object *i* in *N*
- a partial trajectory $T = \langle b_{i1}, b_{i2}, \ldots, b_{il} \rangle$
- a prediction horizon $k > 0$

Output: We want to

predict the next *k* blocks $\langle b_{il+1}, b_{il+2}, \ldots, b_{il+k} \rangle$ of the partial trajectory *T*

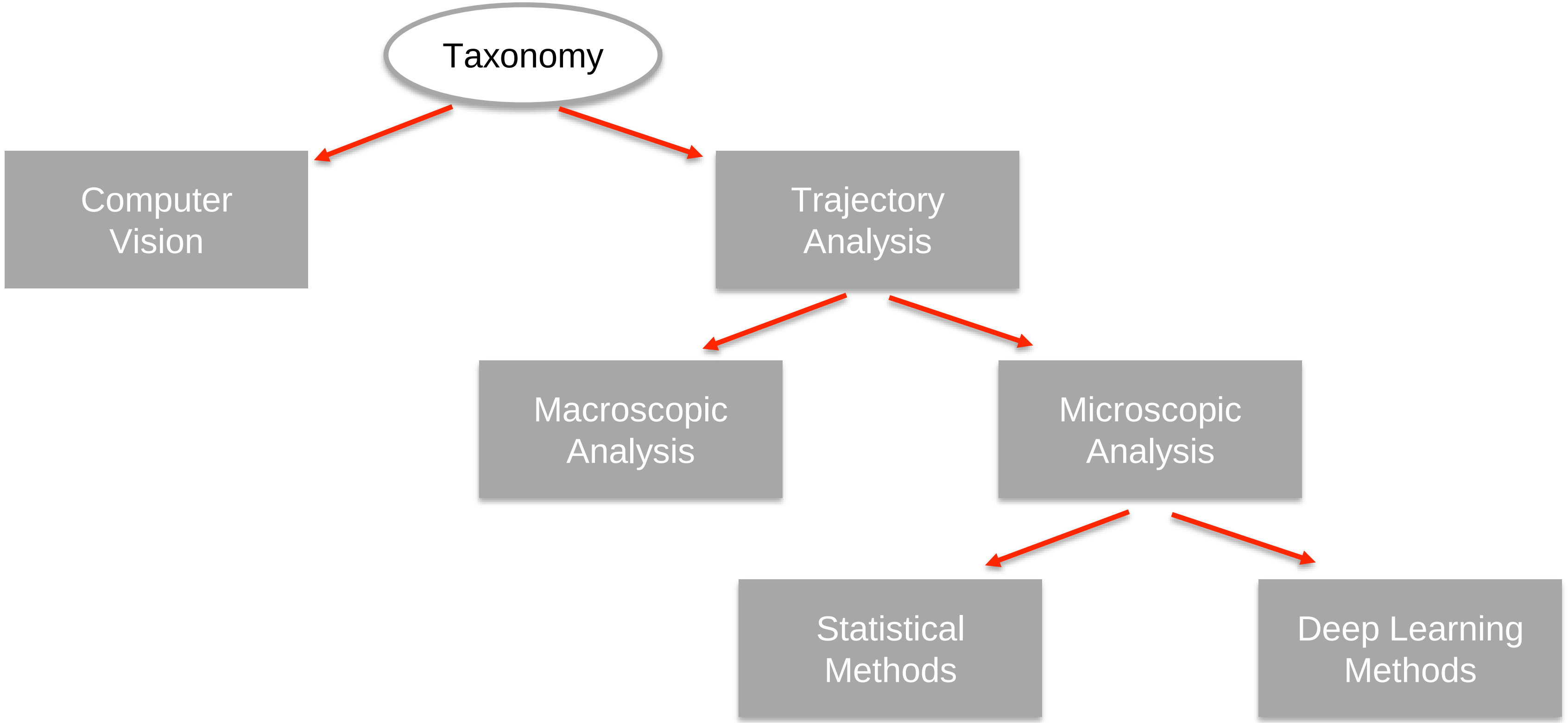
YORK U

Contributions

- **Point2Hex:** GPS trajectories to HO mobility flow data
- Propose to leverage deep generative models for trajectory prediction
- Propose a transformer-based framework **TrajLearn**
- TrajLearn **outperforms** the state-of-the-art baselines
- Make the **source code** publicly available to facilitate the reproducibility

Existing works

Literature Overview



General Related Work

Computer Vision Domain

- Predict future path or movement of objects in a scene (a small scale) over time

Out of the scope: Rely on camera-generated video frames

Macroscopic Trajectory Analysis

- Focus on high-level (city-level or region-level) mobility predictions (instead of individual level)

Different focus: crowd flow prediction [Lin et al. AAAI'19], taxi demand prediction [Yao et al. AAAI'18]

Statistical Methods

Matrix Factorization

- Decompose matrix into matrices that representing object preferences and location attributes

Examples: Fused MF [Cheng et al. AAAI'12], GeoMF [Lian et al. SIGKDD'14], Rank-geofm [Li et al. SIGIR'15]

Markov Chain

- Model the sequence of visits as a chain of states, governed by transition probabilities

Examples: HMM [Mathew et al. UbiComp'12], FPMC-LR [Cheng et al. IJCAI'13], Semantics-aware HMM [Shi et al. TKDE'19]

Limitations

- Limited scalability
- Often rely on assumptions about the data distribution
- Feature engineering is required

Deep Learning Methods - 1/2

RNN/LSTM/GRU

- Use recurrent neural networks to process sequential data

Examples: ST-RNN [Liu et al. AAAI'16], HST-LSTM [Kong et al. IJCAI'18], DeepTrip [Zhang et al. IEEE trans Intell Transp Syst'23]

Attention Mechanism

- Allow models to focus on different parts of the input sequence when producing the output

Examples: DeepMove [Feng et al. WWW'18], GeoSAN [Lian et al. KDD'20], STAN [Luo et al. WWW'21]

Limitations

- Mostly designed for the POI prediction
- Data sparsity and imprecision

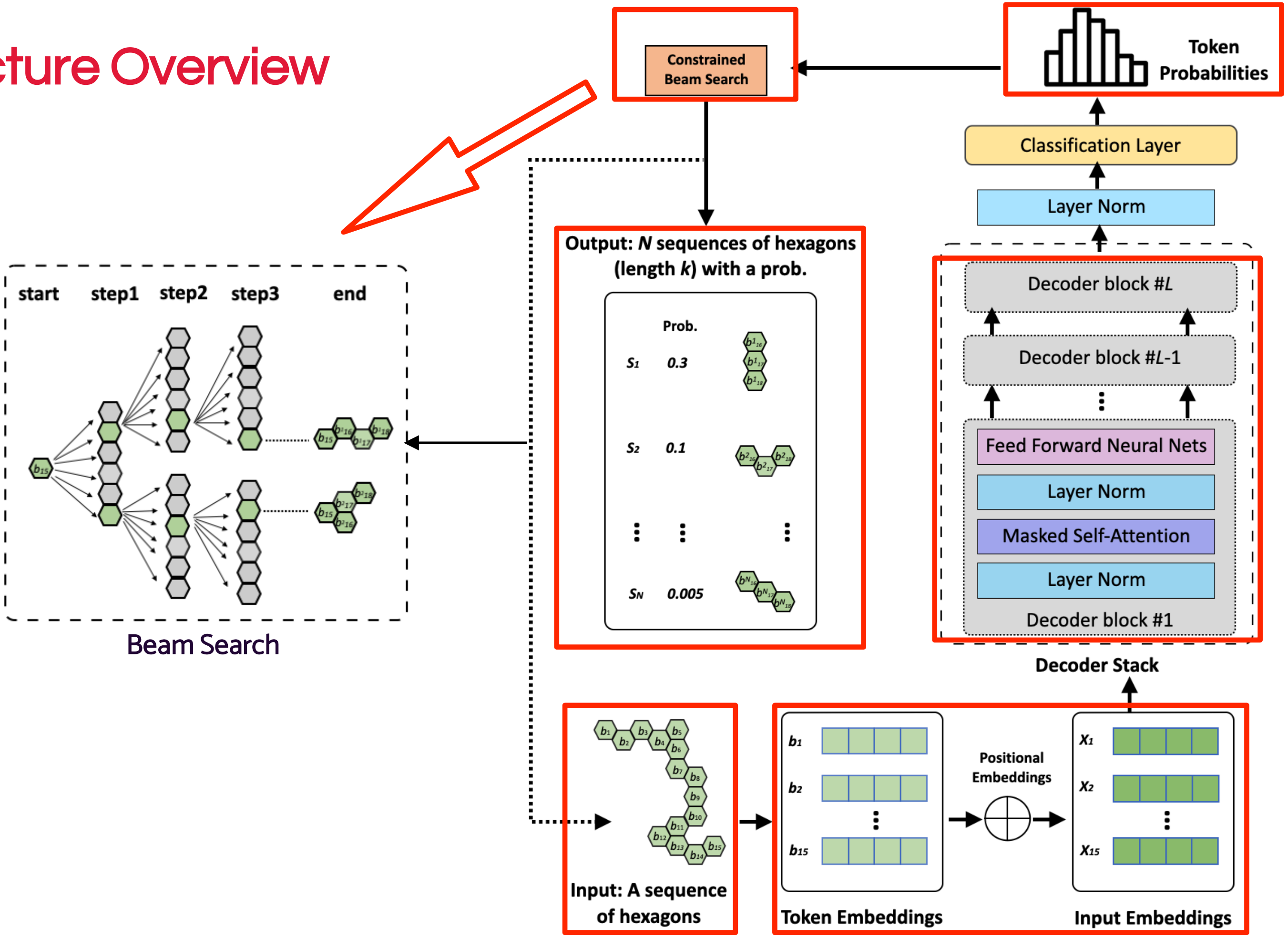
Deep Learning Methods - 2/2

Specialized Works

- DeepUrbanMomentum [Jiang et al. AAAI'18]
  - Limitations: Need other information
- Continuous Trajectory Prediction [Sadri et al. IMWUT'18]
  - Limitations: Heavily rely on a single historical record of an individual
- From movement purpose to mobility prediction [Amichi et al. SIGSPATIAL'21]
  - Limitations: Need to add movement semantic to trajectories

Methodology

Architecture Overview



Input of Transformer

The input to the transformer

$$h_0 = BW_e + W_p$$

Where

: Higher-order mobility flow

: Block embedding matrix

: Position embedding matrix

Hidden State Computation

The hidden state of each transformer layer

$$h_j' = h_{j-1} + Self - Attention(LayerNorm(h_{j-1}))$$

$$h_j = h_j' + FeedForward(LayerNorm(h_j'))$$

Where *LayerNorm()*: Layer normalization

Self-Attention() : Masked multi-head self-attention operation

FeedForward() : Position-wise feed-forward network

YORK U

Activation Function

Gaussian Error Linear Unit (GELU)

$$GELU(x) = x \cdot P(X \leq x)$$

Where and implemented as

$$0.5x\left(1 + \tanh\left(\sqrt{\frac{2}{\pi}}\left(x + 0.044715x^3\right)\right)\right)$$

Next Block/Hexagon Prediction

Based on the probabilities of all possible next blocks

$$P(b_{l+1} \mid B) = \text{softmax}(\text{FeedForward}(\text{LayerNorm}(h_L)))$$

Model Training

<EOT> in Trajectories

- Temporal cutoff: time threshold

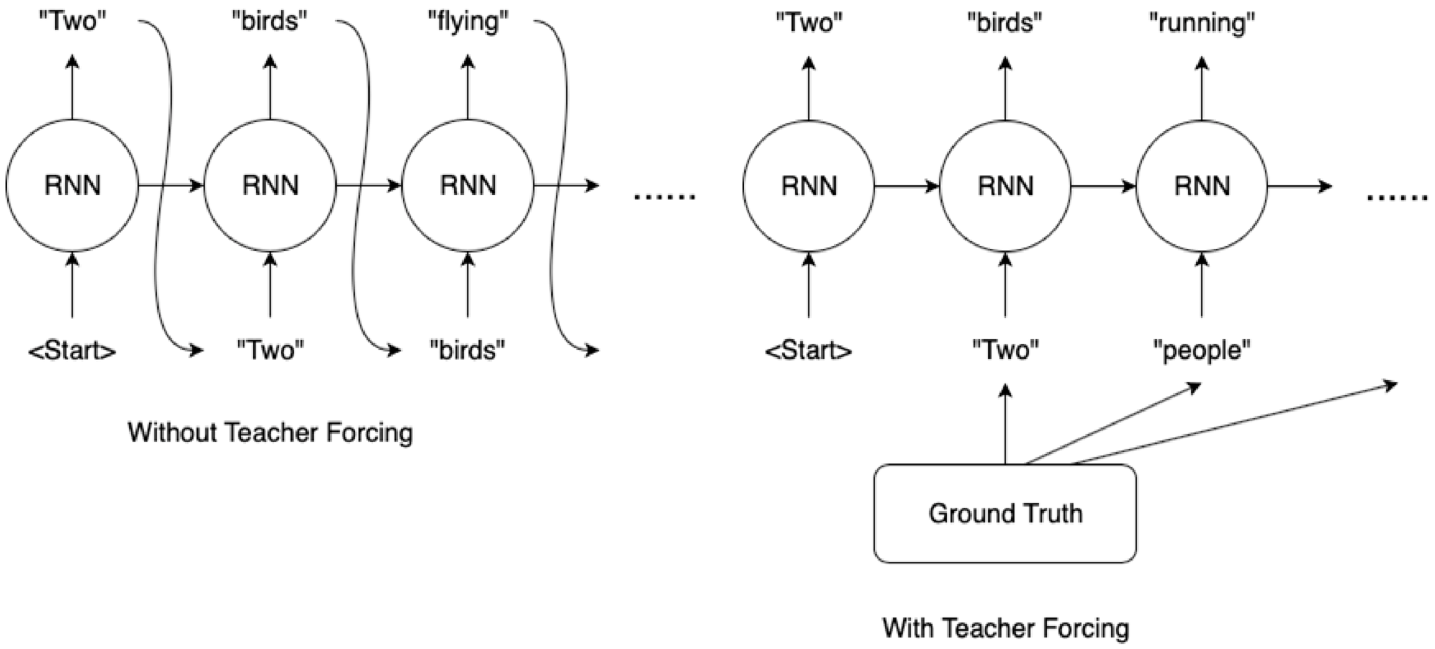
Gap in GPS data beyond this threshold indicates the end of the trajectory

- Spatial cutoff: distance threshold

Distance between consecutive GPS points is greater than this threshold

Model Training

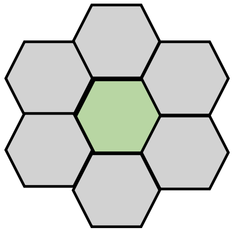
Teaching Forcing

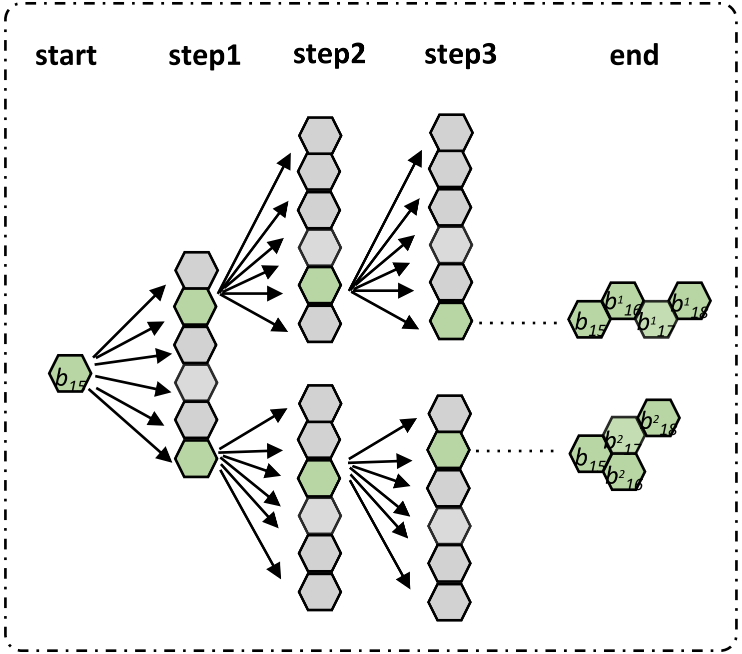


Beam Search with Constraints

A heuristic search algorithm that explores the most promising trajectory paths

- Initialization
- Beam expansion
- Beam pruning
- Termination





Beam Expansion

The probability at each step is updated based on their cumulative probabilities

$$P(b_{i_1}\ldots b_{i_n}) = P(b_{i_1}\ldots b_{i_{n-1}}) \times P(b_{i_n} \mid b_{i_1}\ldots b_{i_{n-1}})$$

Evaluation

Experimental Scenarios

RQ 1) Accuracy

- What is the accuracy performance of our method against baselines?

RQ 2) Sensitivity Analysis

- How does the performance vary with different input trajectory lengths and prediction lengths?

RQ 3) Map Resolution Analysis

- How does the performance vary with different tessellation levels?

RQ 4) Ablation Study

- How does beam search with the constraints impact the performance?

Datasets

Timely ordered trajectory data set is split into:

70% Training, 10 % Validation, 20% Testing

Dataset	Objects	Trajectories	Time Period	Resolutions
HO-Rome	315	5,873	02/01/14 - 03/02/14	{7, 8, 9}
HO-Porto	442	1,668,859	07/01/13 - 06/30/14	{7, 8, 9}
HO-GeoLife	57	2,100	04/01/07 - 10/31/11	{7, 8, 9}

Experimental Setup

Computational Environment

- NVIDIA RTX A6000 graphics card and 320GB of memory
- Implementation: Python 3, PyTorch 1.13

Map Tessellation and Resolutions

- H3 geo-indexing system

Deep Generative Model

- Based on the GPT-2 LLM architecture

Training Parameters

- AdamW optimizer with learning rate = 5×10^{-3}
- Batch size = 64
- Dropout rate = 0.1

Baselines

Statistical Methods

- MC

Deep Learning Methods

- LSTM
- GRU
- LSTM-ATTN
- DeepMove

Our Method

- TrajLearn

Metrics

Accuracy@N [↑]

- Measure the proportion of true samples included in the predictions

$$Accuracy@N = \frac{\left|\{s \mid s \in P, true(s) \in Top_n(s)\}\right|}{|P|}$$

BLEU Score [↑]

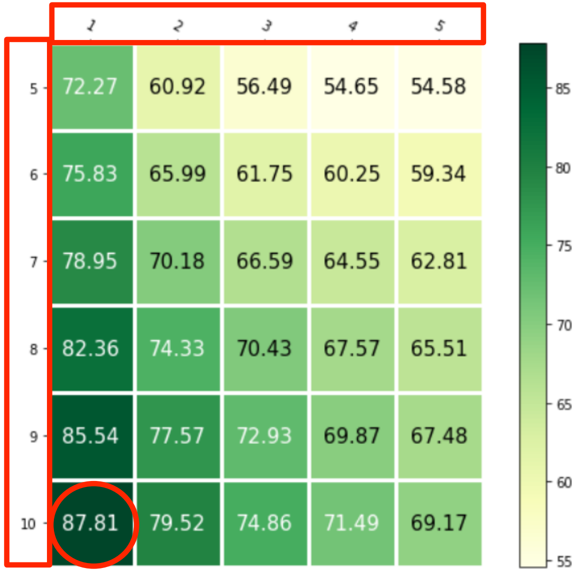
- Measure how many n-grams of the predicted sequence match with the n-grams in the actual sequence

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \qquad BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

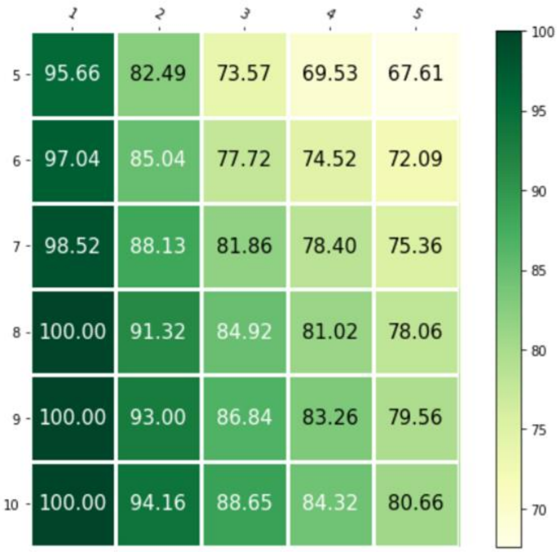
RQ 1) Model Accuracy Performance

		Resolution 7				Resolution 8				Resolution 9			
Dataset	Model	Acc@1	Acc@3	Acc@5	BLEU	Acc@1	Acc@3	Acc@5	BLEU	Acc@1	Acc@3	Acc@5	BLEU
Ho-Porto	MC	0.3284	0.4586	0.4908	0.2444	0.2478	0.3354	0.3893	0.2359	OOM	OOM	OOM	OOM
	LSTM	0.5970	0.6318	0.6400	0.6302	0.4579	0.5087	0.5172	0.5021	0.5044	0.5588	0.5643	0.5479
	LSTM-ATTN	0.1113	0.1923	0.2065	0.2035	0.1112	0.1705	0.1929	0.2065	0.2716	0.3682	0.4011	0.3842
	GRU	0.5532	0.5877	0.5957	0.5866	0.3154	0.3542	0.3606	0.3530	0.3649	0.4086	0.4144	0.4058
	DeepMove	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	TrajLearn (ours)	**0.6917**	**0.8066**	**0.8490**	**0.7691**	**0.5135**	**0.6931**	**0.7590**	**0.5918**	**0.5772**	**0.8022**	**0.8741**	**0.6379**
	Improvement (%)	15.86 %	27.65 %	32.64 %	22.04 %	12.14 %	36.25 %	46.75 %	17.86 %	14.43 %	43.56 %	54.90 %	16.43 %
Ho-Rome	MC	0.2088	0.3982	0.4690	0.0685	0.2374	0.3811	0.4590	0.1504	0.2100	0.3157	0.3564	0.1686
	LSTM	0.2820	0.3138	0.3227	0.3179	0.3932	0.4340	0.4407	0.4315	0.4617	0.5144	0.5186	0.5036
	LSTM-ATTN	0.1079	0.1522	0.1850	0.1819	0.2264	0.2845	0.3055	0.2998	0.2890	0.3704	0.3892	0.3735
	GRU	0.2966	0.3298	0.3385	0.3335	0.3997	0.4400	0.4468	0.4379	0.4638	0.5158	0.5199	0.5052
	DeepMove	0.3406	**0.4969**	**0.5793**	0.2821	0.3860	0.5036	0.5657	0.3286	OOM	OOM	OOM	OOM
	TrajLearn (ours)	**0.3746**	0.4740	0.5167	**0.4215**	**0.4974**	**0.6428**	**0.6996**	**0.5434**	**0.5671**	**0.7657**	**0.8431**	**0.6138**
	Improvement (%)	9.98 %	-4.61 %	-10.81 %	26.38 %	24.44 %	27.64 %	23.67 %	24.09 %	22.27 %	48.45 %	62.17 %	21.49 %
Ho-GeoLife	MC	0.2153	0.4917	0.6050	0.1113	0.2149	0.3951	0.4897	0.0866	0.2063	0.3314	0.3859	0.0848
	LSTM	0.5900	0.6086	0.6114	0.6117	0.5616	0.5836	0.5864	0.5838	0.5725	0.6057	0.6085	0.6039
	LSTM-ATTN	0.4944	0.5559	0.5621	0.5478	0.3496	0.4148	0.4249	0.4101	0.2905	0.3664	0.3959	0.3872
	GRU	0.6229	0.6435	0.6465	0.6439	0.5514	0.5742	0.5779	0.5732	**0.5799**	0.6132	0.6158	**0.6111**
	DeepMove	0.5295	0.6742	0.7370	0.3653	0.4529	0.5699	0.6374	0.3374	OOM	OOM	OOM	OOM
	TrajLearn (ours)	**0.7481**	**0.8247**	**0.8635**	**0.7785**	**0.6249**	**0.7404**	**0.7823**	**0.6558**	0.5664	**0.6781**	**0.7194**	0.6004
	Improvement (%)	20.10 %	22.32 %	17.16 %	20.90 %	11.27 %	26.87 %	22.73 %	12.33 %	-2.32 %	10.58 %	16.82 %	-1.77 %

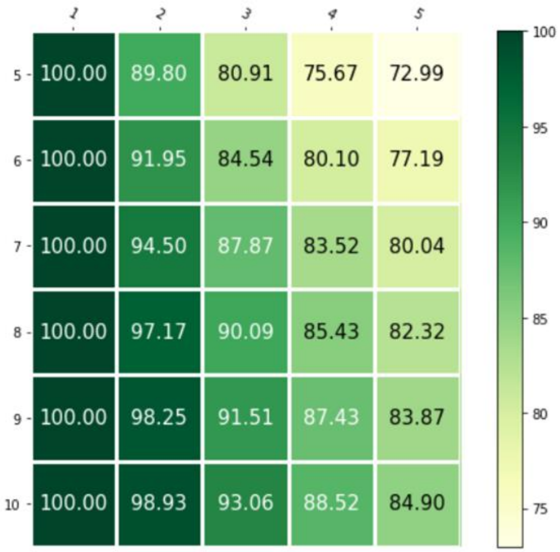
RQ 2) Parameter Sensitivity Analysis



(a) Accuracy@1



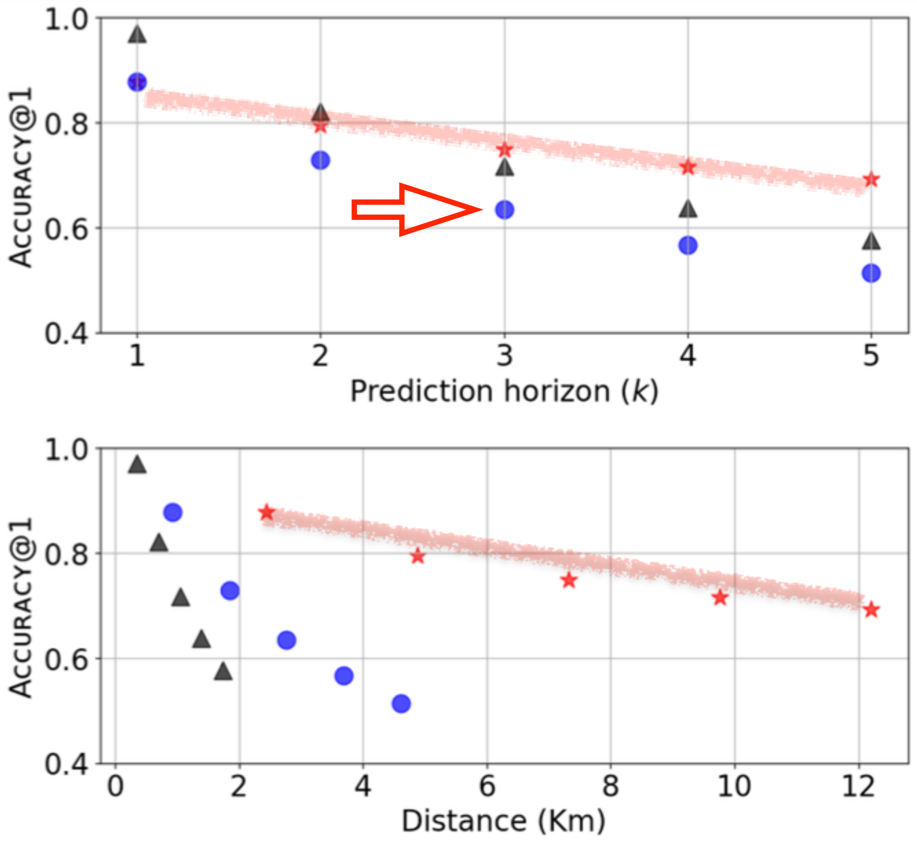
(b) Accuracy@3



(c) Accuracy@5

YORK U

RQ 3) Map Resolution Analysis



7: star

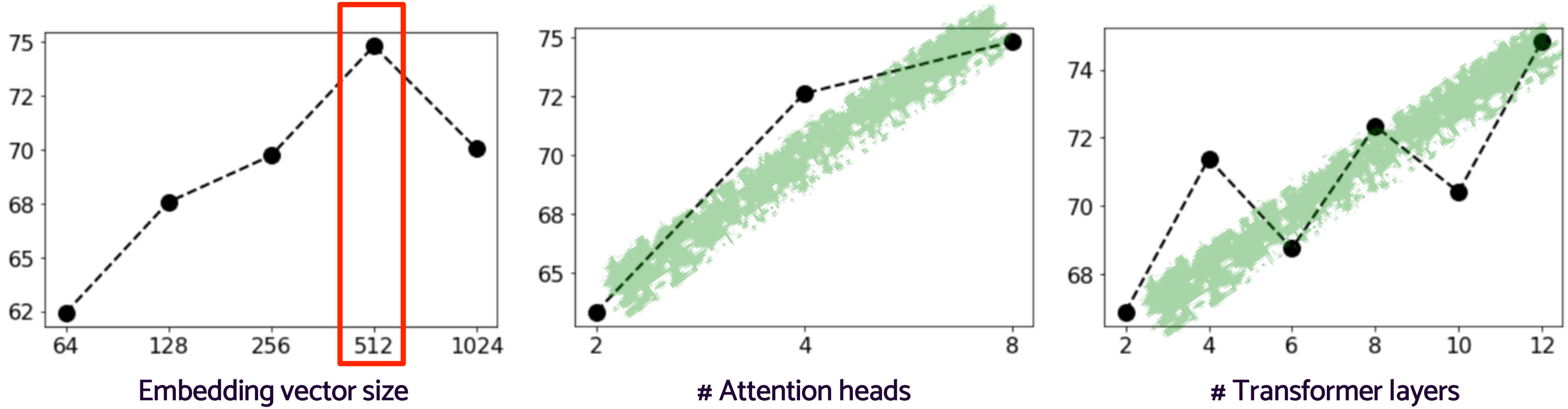
8: circle

9: triangle

YORK U

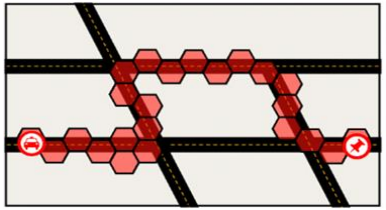
RQ 4) Ablation Study

Dataset	Accuracy@1	Change
Ho-Porto@7	0.6844	-1.07%
Ho-Porto@8	0.4992	-2.86%
Ho-Porto@9	0.5672	-1.76%



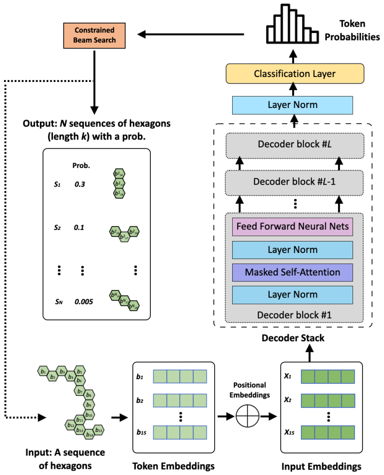
Conclusions

Summary

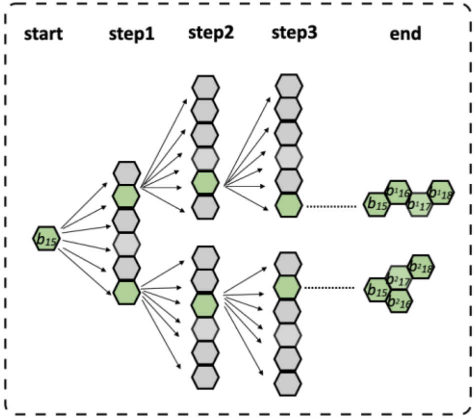


point2hex: software and datasets

GenAI for trajectory prediction



TrajLearn



Beam search

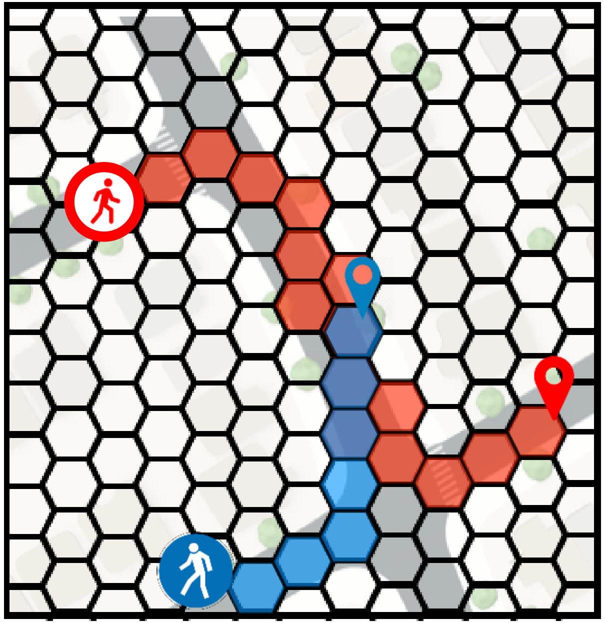
YORK U

Limitations

- Discretization and precision:
  - Too coarse → miss important details
  - Too fine → increased computational complexity
- Data volume
  - May end up with a large amount of data → strain computational resources and require efficient data storage

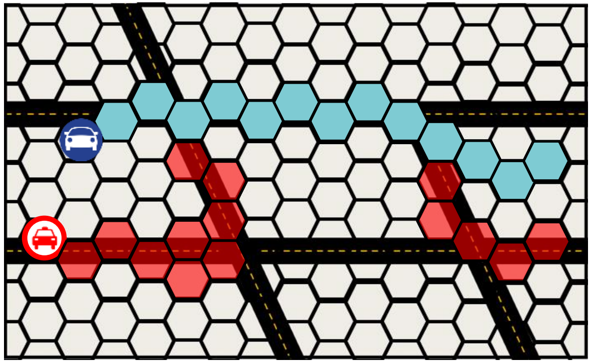
Future Work - Interaction Prediction

Can we use trajectory prediction models for predicting mobility network interactions?

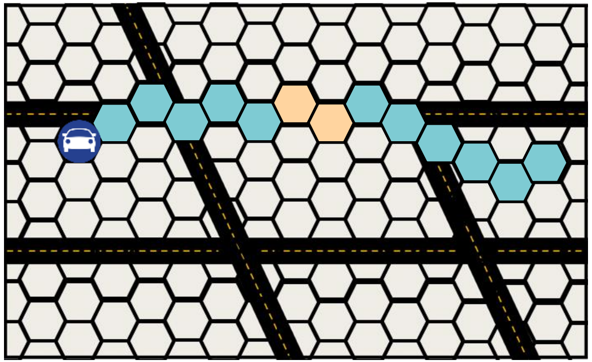


Future Work - Trajectory Foundation Model

Can we develop trajectory foundation models for addressing many trajectory-related tasks?



Trajectory similarity



Trajectory imputation

Papers Published/Submitted

- (Big Data Research) T. Pechlivanoglou, J. Li, J. Sun, F. Heidari, M. Papagelis, "**Epidemic Spreading in Trajectory Networks**", Vol. 27, 100275, pp 1-15, 2022
- (ACM SIGSPATIAL) T. Pechlivanoglou, G. Alix, N. Yanin, J. Li, F. Heidari, and M. Papagelis, "**Microscopic modeling of spatiotemporal epidemic dynamics**", pp 11–21, 2022
- (IEEE MDM) G. Alix, N. Yanin, T. Pechlivanoglou, J. Li, F. Heidari and M. Papagelis, "**A Mobility-based Recommendation System for Mitigating the Risk of Infection during Epidemics**", pp 292-295, 2022
- (ACM SIGSPATIAL) A. Faraji*, J. Li*, G. Alix, M. Alsaeed, N. Yanin, A. Nadiri, and M. Papagelis, "**Point2Hex: Higherorder Mobility Flow Data and Resources**", pp 1-12, 2023
- (Submitted) A. Nadiri, A. Faraji, J. Li, and M. Papagelis, "**TrajLearn: Leveraging Generative Models for Trajectory Prediction Learning,**" pp 1-10

Thank you!

Questions?