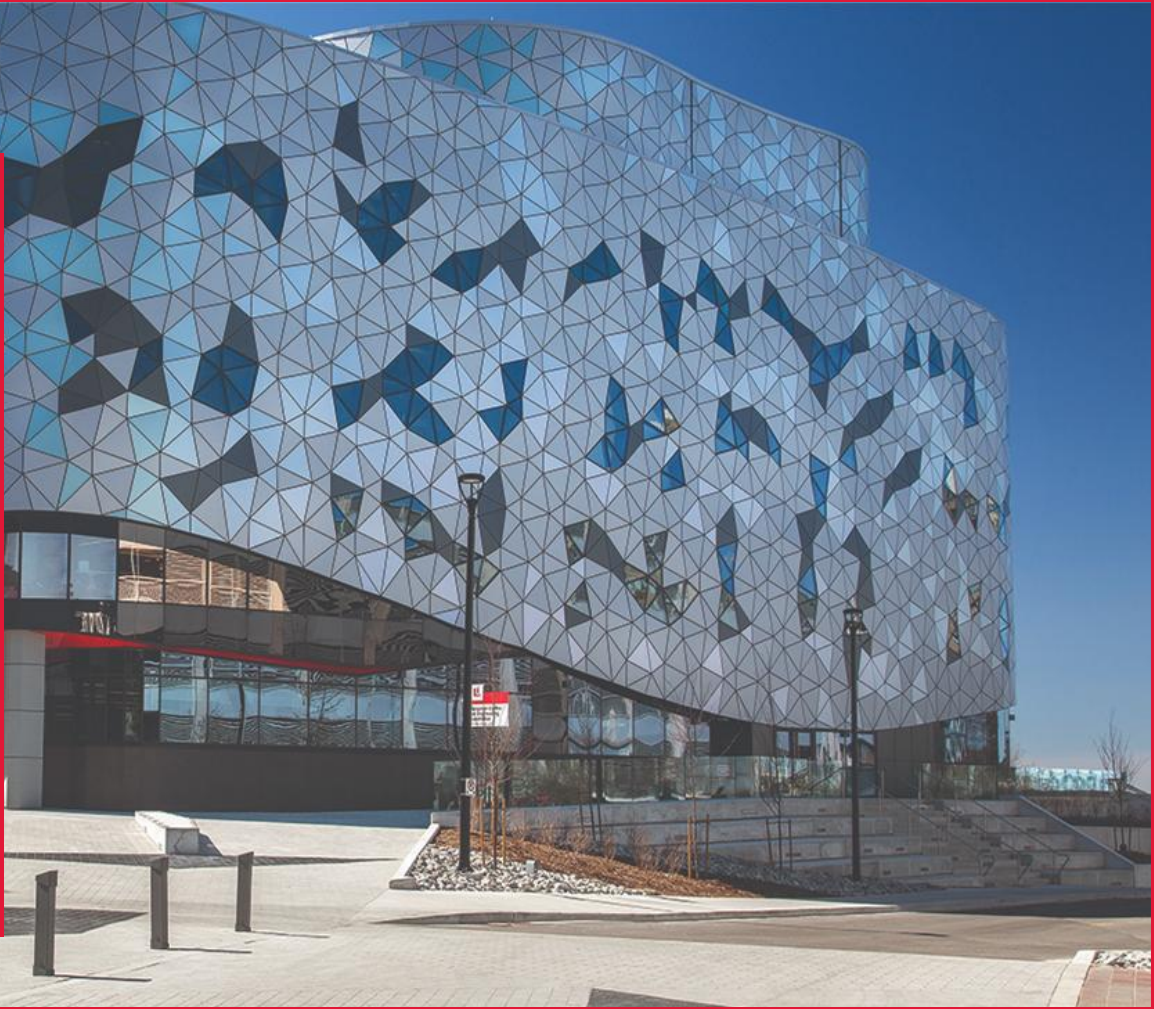
Deep Generative Models for Trajectory Prediction and Mobility Network Forecasting

MSc. Thesis of Amirhossein Nadiri
Department of Electrical Engineering and Computer Science

Jun 11, 2025

YORK U



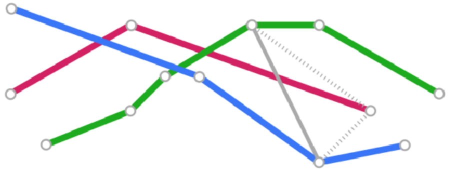
Introduction

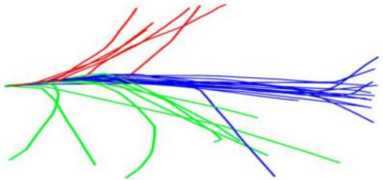
Trajectory/Mobility Data

Trajectory: A Sequence of (Spatiotemporal) Points

Vast Amounts of Trajectory/Mobility Data

Trajectory-related Problems

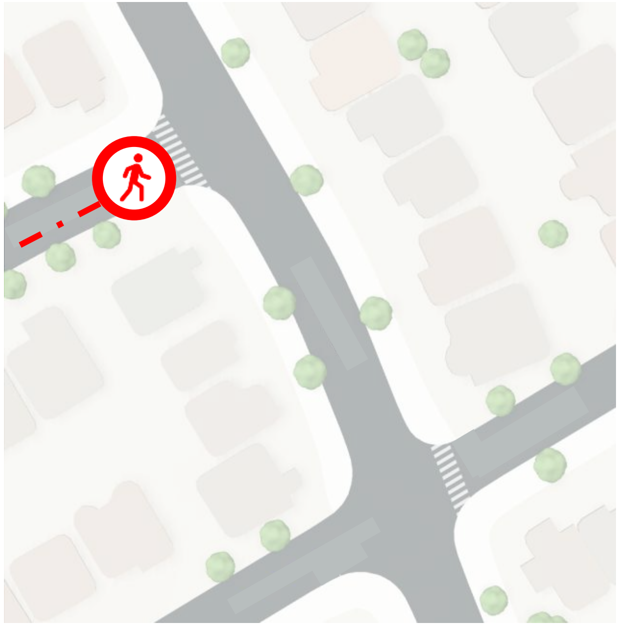




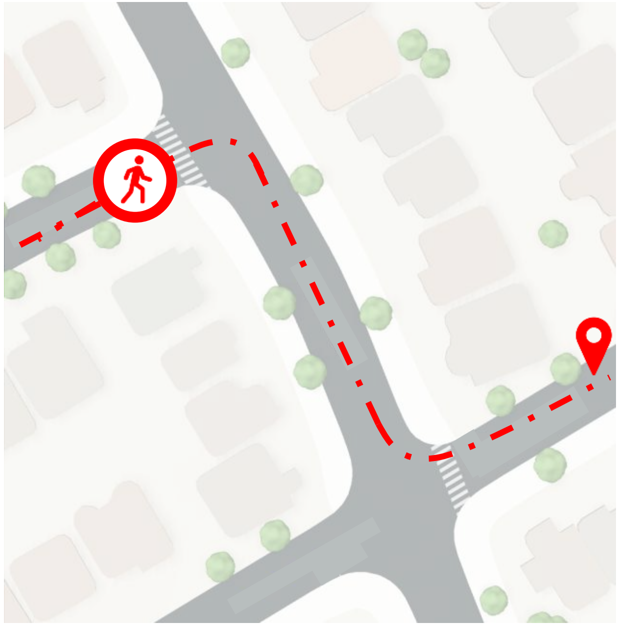
trajectory similarity
trajectory clustering
trajectory imputation
pedestrian crowd behavior

...

Problem 1: Trajectory Prediction

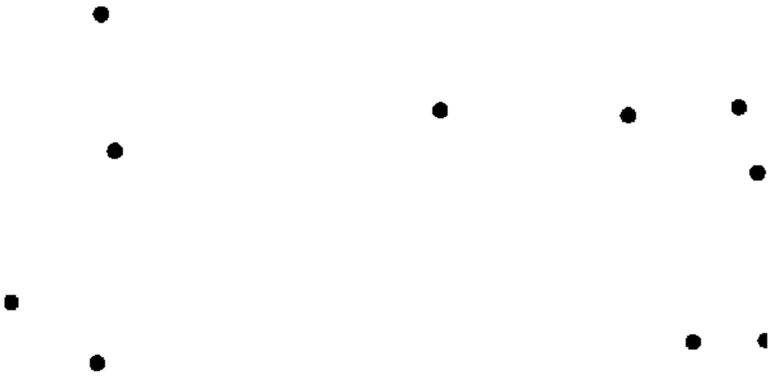


History trajectory

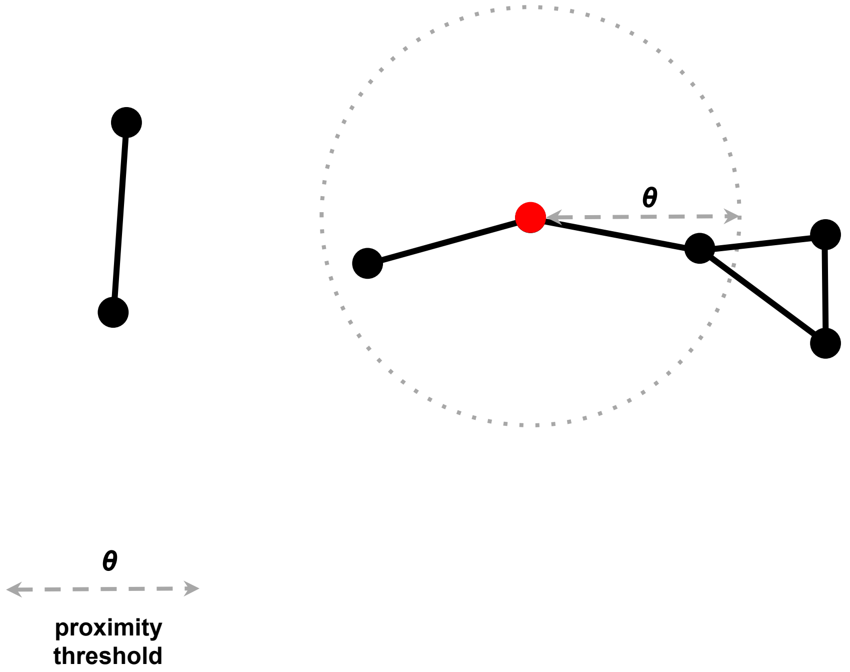


Predict future contact

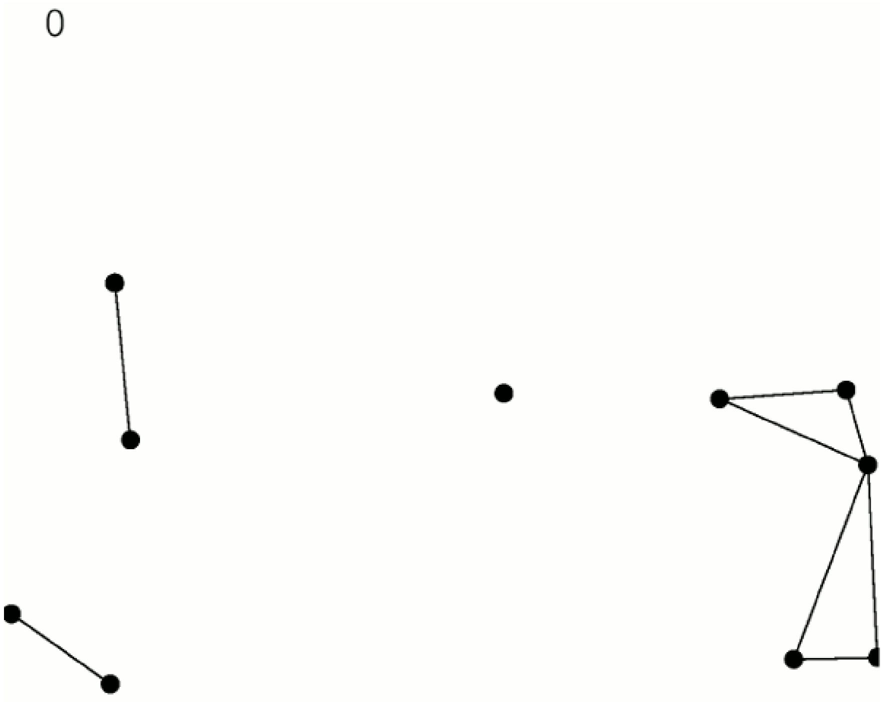
Trajectories of moving objects



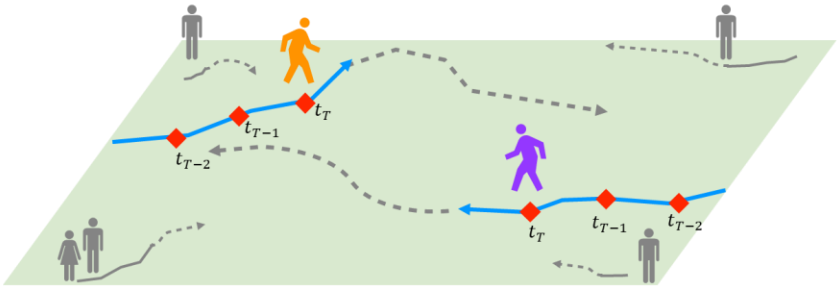
Proximity Network

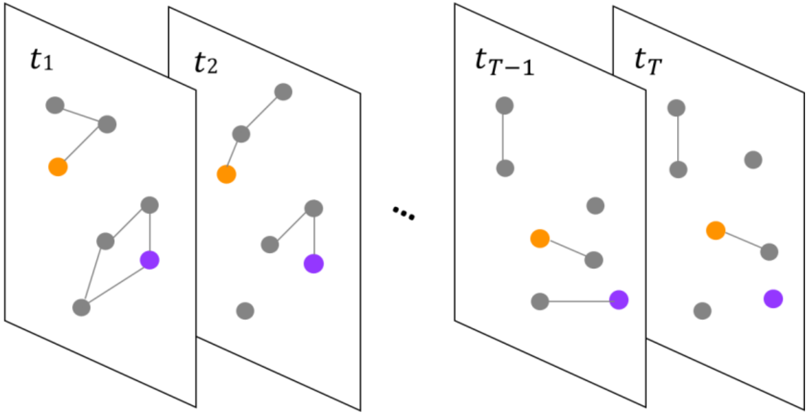


Mobility Network



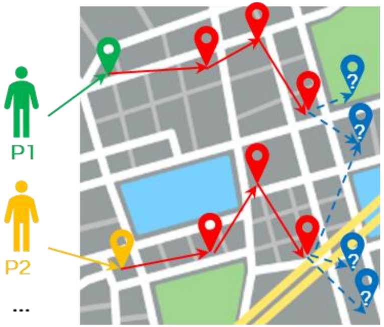
Problem 2: Mobility Network Prediction





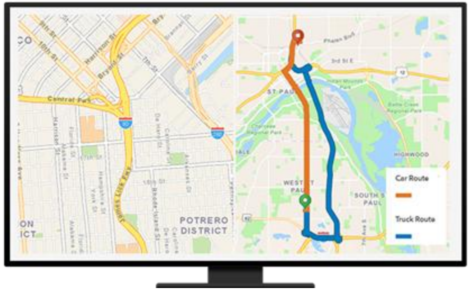
Plethora of Applications

Ride-sharing services



Next POI recommendation

Autonomous vehicles



Traffic flow optimization

Problem Statement

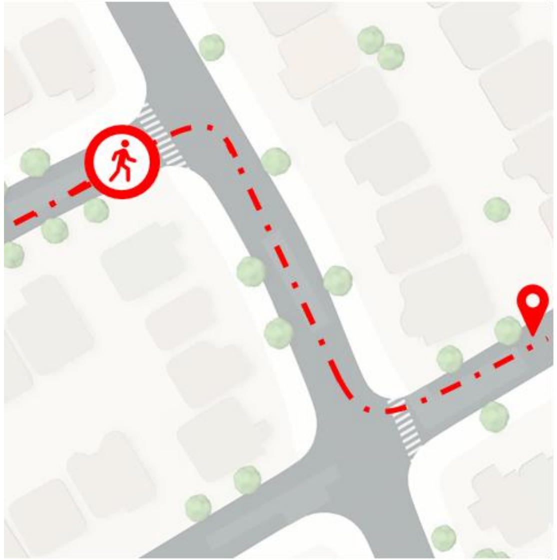
Trajectory Prediction

Let

- an observation area $\mathcal{M}$
- a set of objects $\mathcal{N}$ and their history trajectories $\mathcal{T}^l$
- an observation period $[0, \mathcal{W}]$

Input: Given

- a moving object i in $\mathcal{N}$
- a partial trajectory $T = < p_{i_1}, p_{i_2}, \dots, p_{i_l} >$
- a prediction horizon $k > 0$



Output: We want to

predict the next k spatiotemporal points $< p_{i_{l+1}}, p_{i_{l+2}}, \dots, p_{i_{l+k}} >$ of the partial trajectory T

YORK U

Mobility Network Prediction

Let

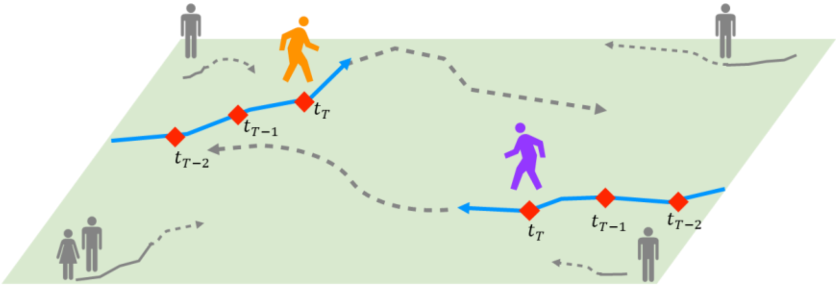
- an observation area $\mathcal{M}$
- a set of objects $\mathcal{N}$ and their history trajectories $\mathcal{T}^l$
- an observation period $[0, \mathcal{W}]$

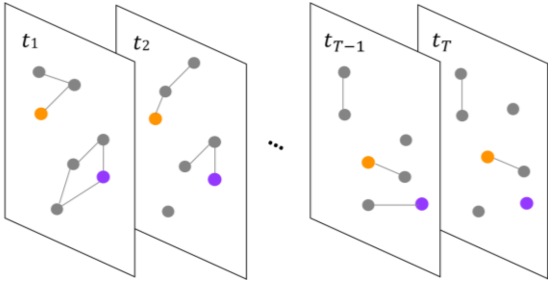
Input: Given

- a prediction horizon $k > 0$

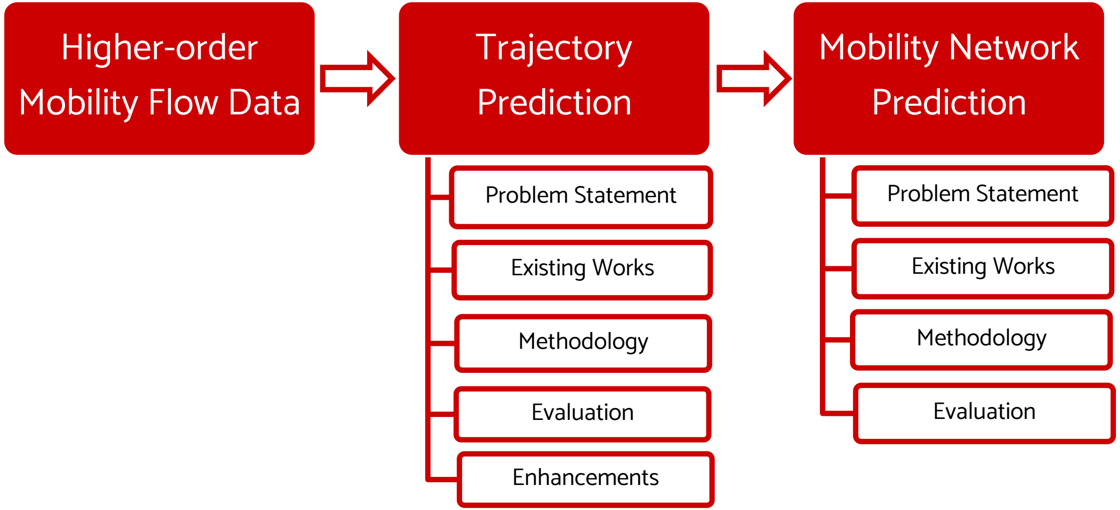
Output: We want to

predict the next k future proximity networks $< G_{i_{w+1}}, G_{i_{w+2}}, \ldots, G_{i_{w+k}} >$





Overview


Higher-order Mobility Flow Data
Trajectory Prediction
Problem Statement
Existing Works
Methodology
Evaluation
Enhancements
Mobility Network Prediction
Problem Statement
Existing Works
Methodology
Evaluation

Higher-order Mobility Flow Data

Challenges of Working with Trajectory Data

Data Sparsity

Limited data

DS

Model Incompatibility

Not compatible with well-known machine learning models

MI

ID

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

Imbalanced Data

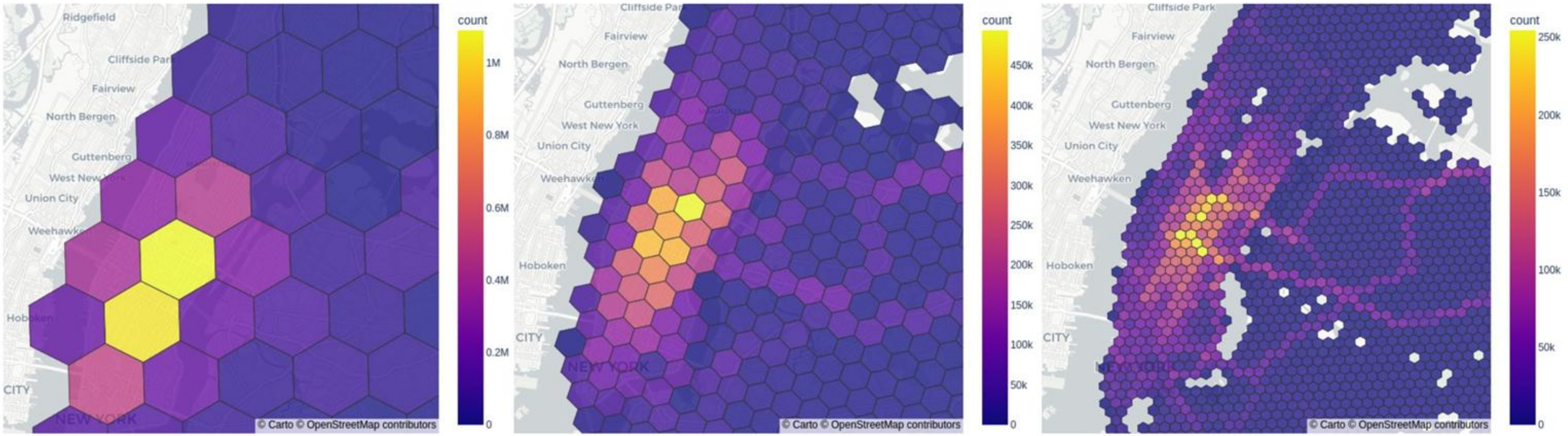
DQ

Low accuracy and completeness

Data Quality

Map Tessellation

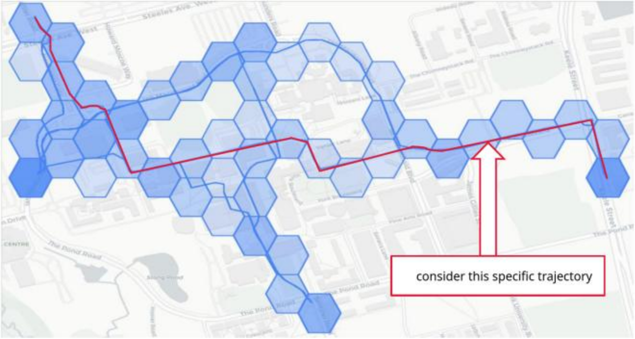
Low resolution → High resolution

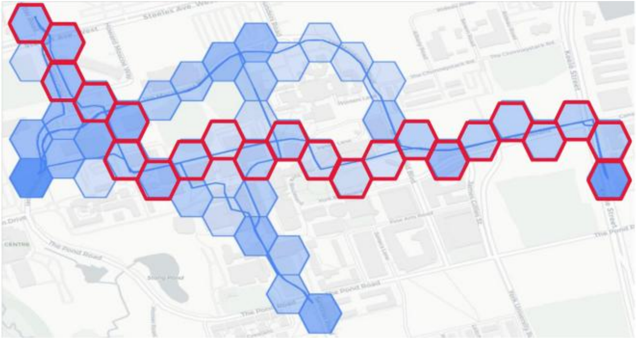


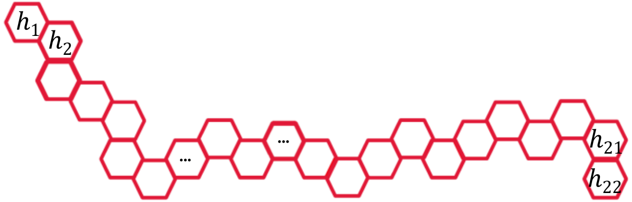
Why hexagons?

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

Trajectories: Sequences of Hexagons



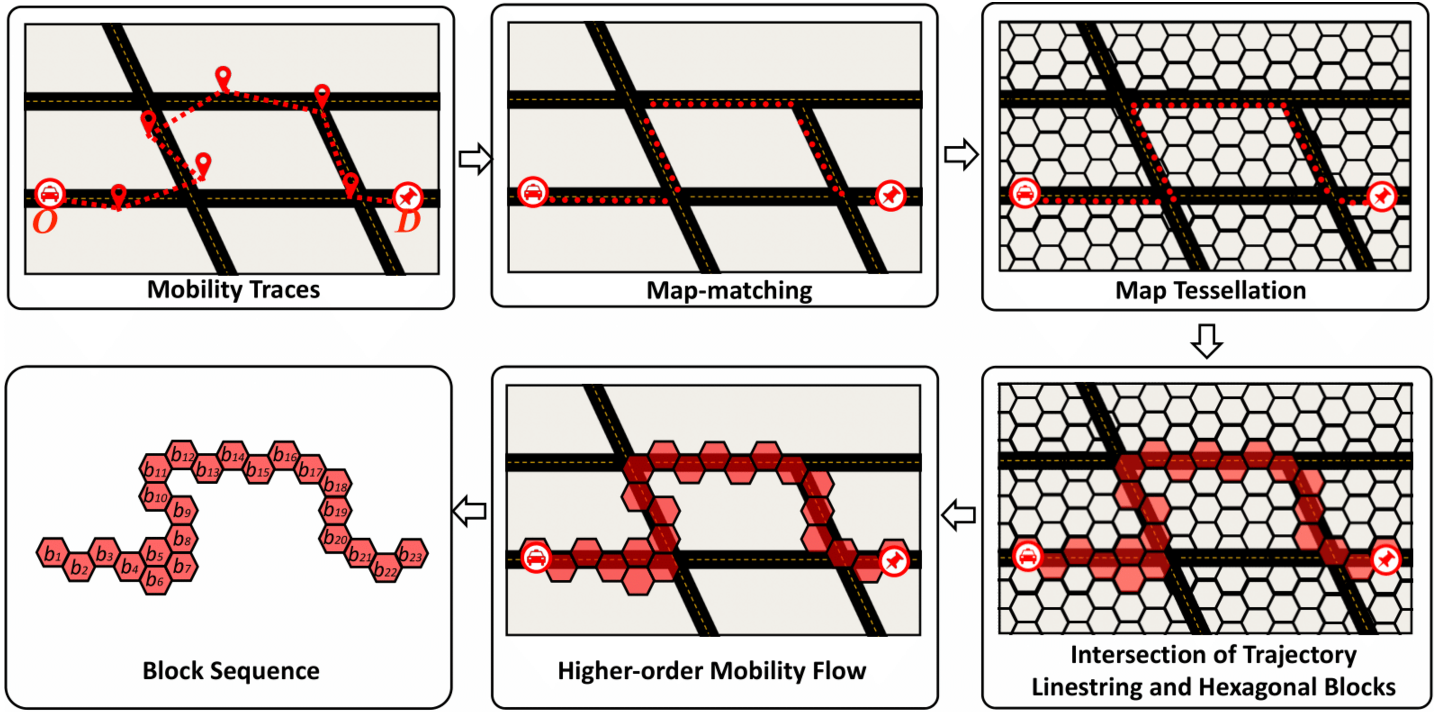




Trajectory: $h_1, h_2, h_3 \; \ldots \; h_{20}, h_{21}, h_{22}$

YORK U

Overview of the Pipeline



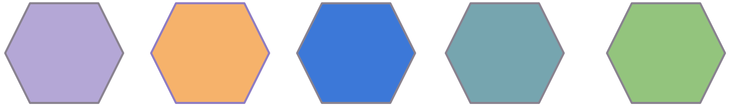
Treat Trajectories as Language Statement

YORK U

Treat Trajectories as Language Statements

Hexagons represent 'tokens' & trajectories represent 'sentences'

Trajectory:



Sentence: I like to learn English

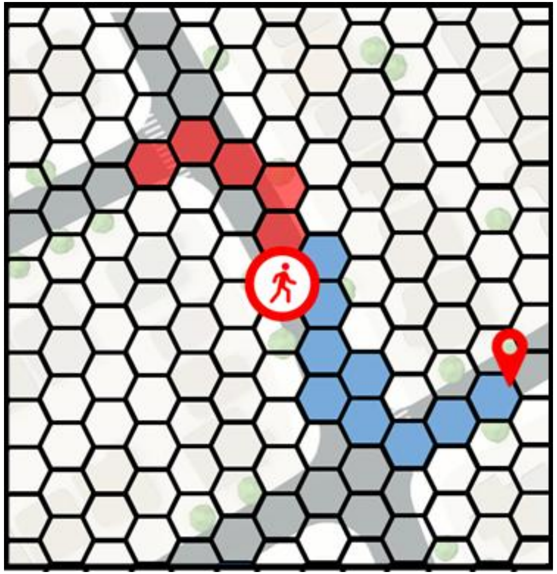
Advantages:

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

YORK U

Problem 1: Trajectory Prediction

Trajectory Prediction (Revisited)



Let

- an observation area $\mathcal{M}$
- a set of objects $\mathcal{N}$ and their history trajectories $\mathcal{T}^l$
- an observation period $[0, \mathcal{W}]$

Input: Given

- a moving object i in $\mathcal{N}$
- a partial trajectory $T =< b_{i_1}, b_{i_2}, \dots, b_{i_l} >$
- a prediction horizon $k > 0$

Output: We want to

predict the next k blocks $< b_{i_{l+1}}, b_{i_{l+2}}, \dots, b_{i_{l+k}} >$ of the partial trajectory T

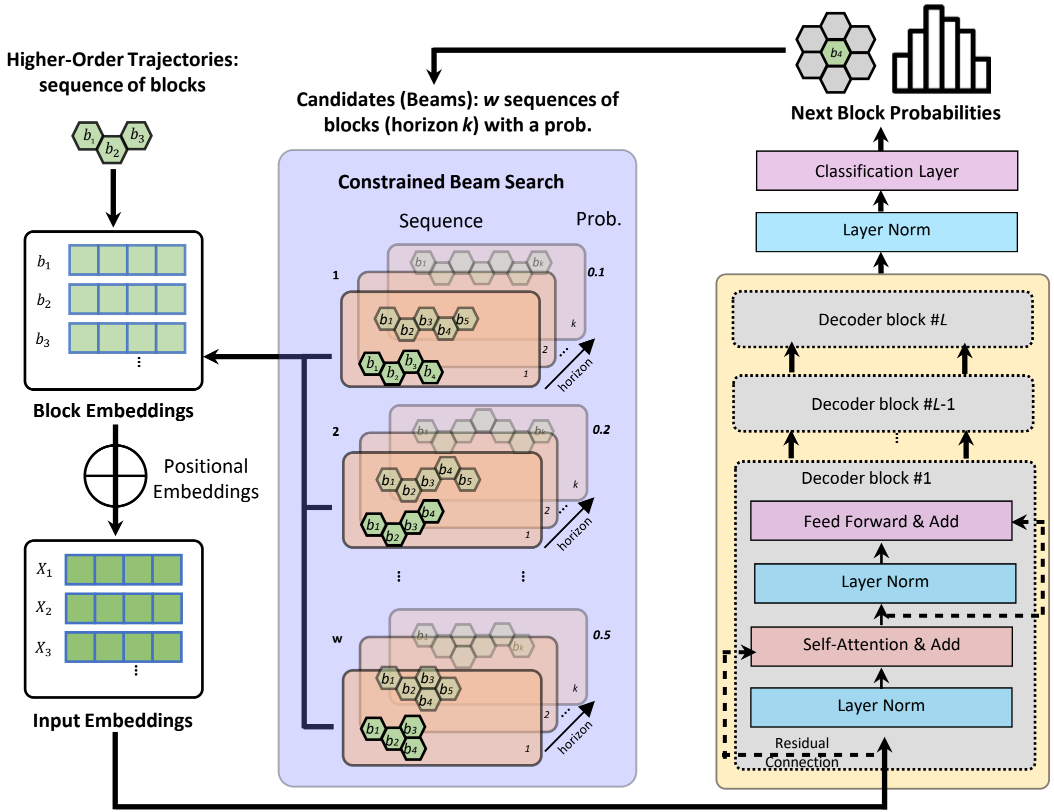
Existing works

Literature Overview

- Trajectory Prediction
  - Computer Vision
    - Image
    - Video
  - Trajectory Analysis
    - GPS data
    - Statistical Methods
      - Matrix Factorization
      - Markov Chain
    - Deep Learning Methods
      - LSTM
      - GRU
      - DeepMove
      - FlashBack

Methodology

TrajLearn Overview



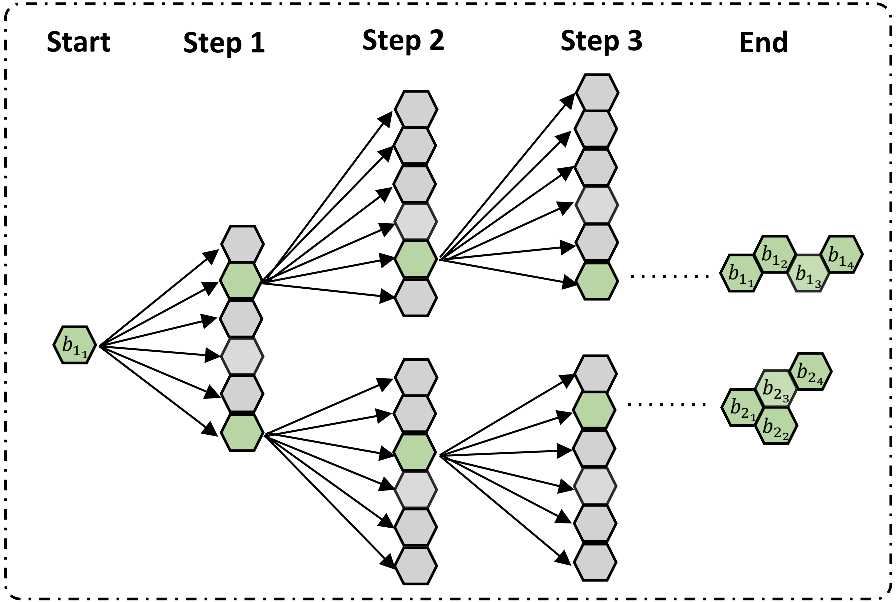
Beam Search with Constraints

A heuristic search algorithm that explores the most promising trajectory paths

$$P(b_{i_1} b_{i_2}) = P(b_{i_1}) \times P(b_{i_2} | b_{i_1})$$

$$P(b_{i_1} b_{i_2} b_{i_3}) = P(b_{i_1} b_{i_2}) \times P(b_{i_3} | b_{i_1} b_{i_2})$$

$$P(b_{i_1} b_{i_2} b_{i_3} b_{i_4}) = P(b_{i_1} b_{i_2} b_{i_3}) \times P(b_{i_4} | b_{i_1} b_{i_2} b_{i_3})$$



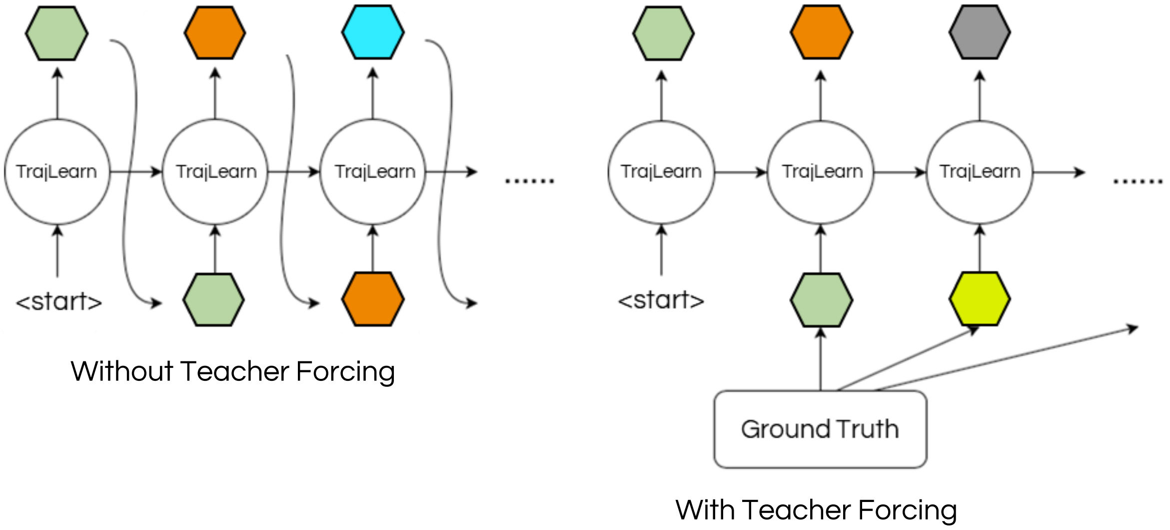
Model Training

<EOT> in Trajectories

- Temporal cutoff and Spatial cutoff

Implementation technique

- Teacher forcing



Evaluation

Experimental Scenarios

RQ 1) Model Accuracy Performance

- What is the accuracy performance of TrajLearn against baselines?

RQ 2) Parameter Sensitivity Analysis

- How performance varies with different input length and prediction horizon?

RQ 3) Beam Search Analysis

- How is the performance affected by varying values of beam width?

RQ 4) Map Resolution Analysis

- How does the performance change with varying levels of map tessellation?

RQ 5) Ablation Study

- How does beam search model hyperparameters impact performance?

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 resolutions 7 to 9

Training Parameters

- AdamW optimizer with learning rate = 5×10^{-3} to 5×10^{-7},
- Batch size = 64
- Dropout rate = 0.1

YORK U

Datasets

Dataset	Entities	Time Period	Res	#Block	#Trajectory	Avg. Length
			7	172	5,678	72.75
Ho-Rome	315	02/01/14 – 03/02/14	8	875	5,837	260.17
			9	4,231	5,854	689.75
			7	3,491	45,186	25.55
Ho-Porto	442	07/01/13 – 06/30/14	8	12,998	397,367	24.07
			9	45,633	1,151,544	35.33
			7	1,878	1,556	117.58
HO-GeoLife	57	04/01/07 - 10/31/11	8	6,360	1,830	219.28
			9	21,270	1,964	525.72

Timely ordered trajectory data set is split into:

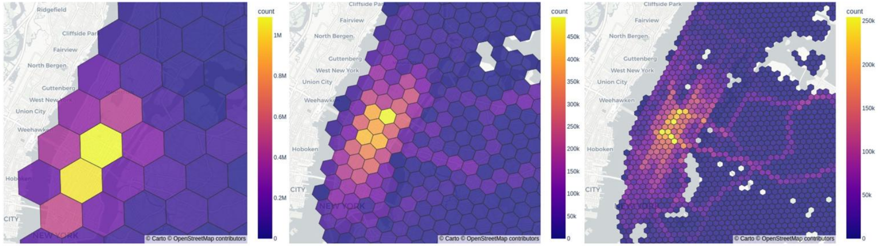
70% Training, 10 % Validation, 20% Testing

Map Tessellation and Resolutions

Resolution	Hex Edge Length (Km)	Hex Area (Km²)	Avg. Length
7	1.406	5.161	72.75
8	0.531	0.737	260.17
9	0.201	0.105	689.75
10	0.076	0.015	25.55

H3 geo-indexing system

7 8 9



Baselines

Statistical Methods

- MC [Gambs et al., MPM'12]

Deep Learning Methods

- LSTM
- GRU
- LSTM-ATTN [Luong et al., EMNLP'15]
- DeepMove [Feng et al., WWW'18]
- Flashback++ [Deng et al., ACM TIST'23]

Our Method

- TrajLearn

Computational complexity

Model	Time Complexity	Memory Complexity	# Parameters
MC	$O(1)$	$O(1)$	≈ 10 K
LSTM	$O(T \cdot H^2)$	$O(T \cdot H)$	≈ 6.1 M
LSTM-ATTN	$O(T \cdot H^2 + T^2 \cdot H)$	$O(T^2 \cdot H)$	≈ 6.1 M
GRU	$O(T \cdot H^2)$	$O(T \cdot H)$	≈ 5.1 M
DeepMove	$O(T \cdot H^2 + T^2 \cdot H)$	$O(T^2 \cdot H)$	≈ 8.4 M
Flashback++	$O(T \cdot H^2)$	$O(T \cdot H)$	≈ 6.5 M
TrajLearn (ours)	$O(T \cdot H^2 + T^2 \cdot H)$	$O(T^2 \cdot H)$	≈ 7.3 M

Inference complexity for single step prediction

H: Hidden embedding dimension

T: Input trajectory length

Metrics

Accuracy@N [↑]

- Measure the proportion of true samples included in the predictions

$$Accuracy@N = \frac{|\{s \in P |\ \mathrm{true}(s) \in \mathrm{Top}_N(s)\}|}{|P|}$$

BLEU Score [↑]

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

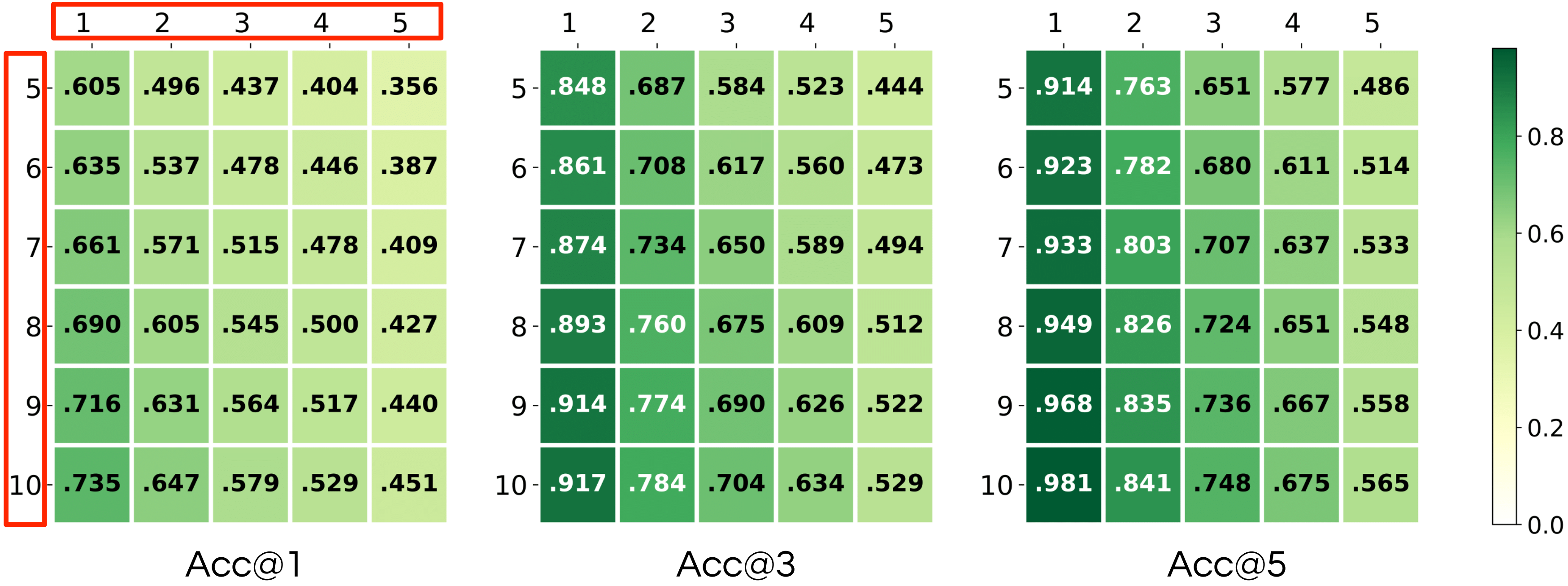
$$BLEU = BP \cdot \sum_{n=1}^{T} w_n \log(p_n)\,, \qquad BP = \begin{cases} 1 & \text{if } c > r \\ e^{1-\frac{r}{c}} & \text{if } c \leq r \end{cases}$$

RQ 1) Model Accuracy Performance

Dataset	Model	Resolution 7				Resolution 8				Resolution 9			
		Acc@1	Acc@3	Acc@5	BLEU	Acc@1	Acc@3	Acc@5	BLEU	Acc@1	Acc@3	Acc@5	BLEU
Ho-Porto	MC	0.2232	0.246	0.2483	0.2443	0.2131	0.239	0.2426	0.2359	0.2239	0.2511	0.2561	0.249
	LSTM	0.436	0.507	0.5325	0.4836	0.3268	0.4435	0.4888	0.3778	0.3744	0.5168	0.5626	0.4132
	LSTM-ATTN	0.4383	0.5112	0.5387	0.4891	0.3233	0.4377	0.4781	0.3719	0.3675	0.5086	0.5556	0.4075
	GRU	0.314	0.3741	0.4089	0.3581	0.201	0.2771	0.3067	0.2411	0.2236	0.3074	0.3364	0.2588
	DeepMove	0.0802	0.2038	0.345	0.1516	0.1272	0.1994	0.2482	0.1725	0.2463	0.3139	0.3497	0.2858
	Flashback++*	0.4439	0.5015	0.5254	0.4929	0.332	0.4599	0.4867	0.4066	0.3993	0.5316	0.5809	0.4314
	TrajLearn (ours)	**0.4507**	**0.5285**	**0.5648**	**0.5108**	**0.4244**	**0.5638**	**0.6138**	**0.486**	**0.4785**	**0.6555**	**0.7111**	**0.5255**
	Improvement	**1.53%**	**5.38%**	**7.5%**	**3.63%**	**27.84%**	**22.59%**	**26.11%**	**19.53%**	**19.81%**	**23.3%**	**22.42%**	**21.8%**
Ho-Rome	MC	0.044	0.0591	0.0643	0.0685	0.1335	0.1482	0.1512	0.1504	0.1459	0.1699	0.1726	0.1686
	LSTM	0.2284	0.2919	0.3195	0.2566	0.3191	0.4152	0.4536	0.349	0.3664	0.502	0.5527	0.3977
	LSTM-ATTN	0.2264	0.2892	0.317	0.255	0.3164	0.4133	0.4508	0.3462	0.3663	0.5016	0.5522	0.3972
	GRU	0.2132	0.2656	0.2934	0.2392	0.2244	0.2806	0.3064	0.248	0.1636	0.242	0.2801	0.1932
	DeepMove	0.2644	0.3594	0.394	0.2966	0.3529	0.414	0.4367	0.3815	OOM	OOM	OOM	OOM
	Flashback++	0.2448	0.3086	0.3386	0.2658	0.3364	0.4292	0.4666	0.3634	0.386	0.5269	0.5821	0.4225
	TrajLearn (ours)	**0.2924**	**0.37**	**0.4046**	**0.3279**	**0.3953**	**0.5149**	**0.5631**	**0.4317**	**0.4515**	**0.6085**	**0.6704**	**0.4886**
	Improvement	**10.6%**	**2.95%**	**2.69%**	**10.56%**	**17.5%**	**19.95%**	**20.7%**	**18.8%**	**16.96%**	**15.51%**	**15.15%**	**15.63%**
Ho-GeoLife	MC	0.1045	0.1083	0.1093	0.1113	0.0743	0.0848	0.0858	0.0864	0.0665	0.0882	0.0899	0.0857
	LSTM	0.3837	0.471	0.5009	0.3996	0.3632	0.4325	0.4631	0.3787	0.3909	0.4898	0.5203	0.4166
	LSTM-ATTN	0.4208	0.484	0.5109	0.4376	0.4081	0.4742	0.5048	0.4271	0.3892	0.4809	0.5136	0.4142
	GRU	0.3051	0.3544	0.3995	0.3183	0.2187	0.3135	0.3656	0.2391	0.107	0.1811	0.2437	0.1308
	DeepMove	0.4212	0.5928	0.6679	0.4765	0.3598	0.5136	0.6255	0.3778	OOM	OOM	OOM	OOM
	Flashback++	0.3907	0.4755	0.5072	0.4072	0.3911	0.442	0.4885	0.3995	0.4144	0.5154	0.5301	0.4311
	TrajLearn (ours)	**0.6008**	**0.6683**	**0.7028**	**0.6235**	**0.5303**	**0.6082**	**0.6427**	**0.5565**	**0.4266**	**0.5247**	**0.5589**	**0.4545**
	Improvement	**42.6%**	**12.75%**	**5.23%**	**30.82%**	**35.6%**	**37.63%**	**31.56%**	**39.3%**	**2.94%**	**1.8%**	**8.44%**	**5.43%**

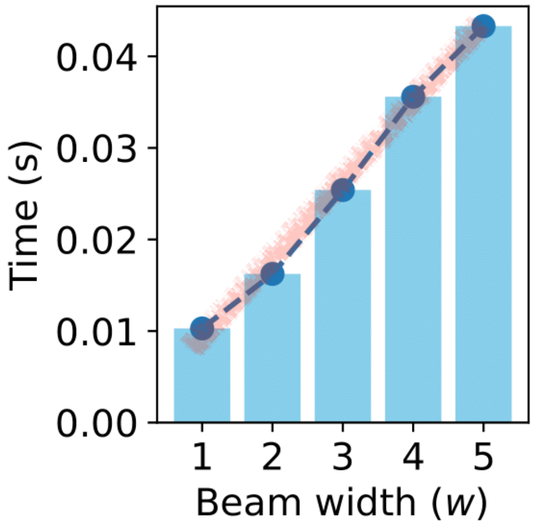
Input length = 10, Prediction horizon = 5

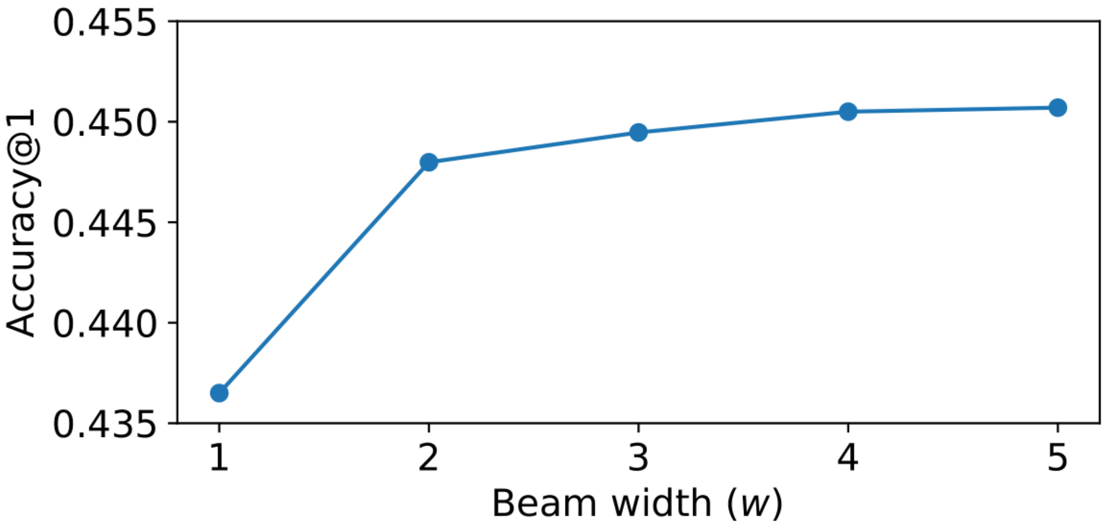
RQ 2) Parameter Sensitivity Analysis



YORK U

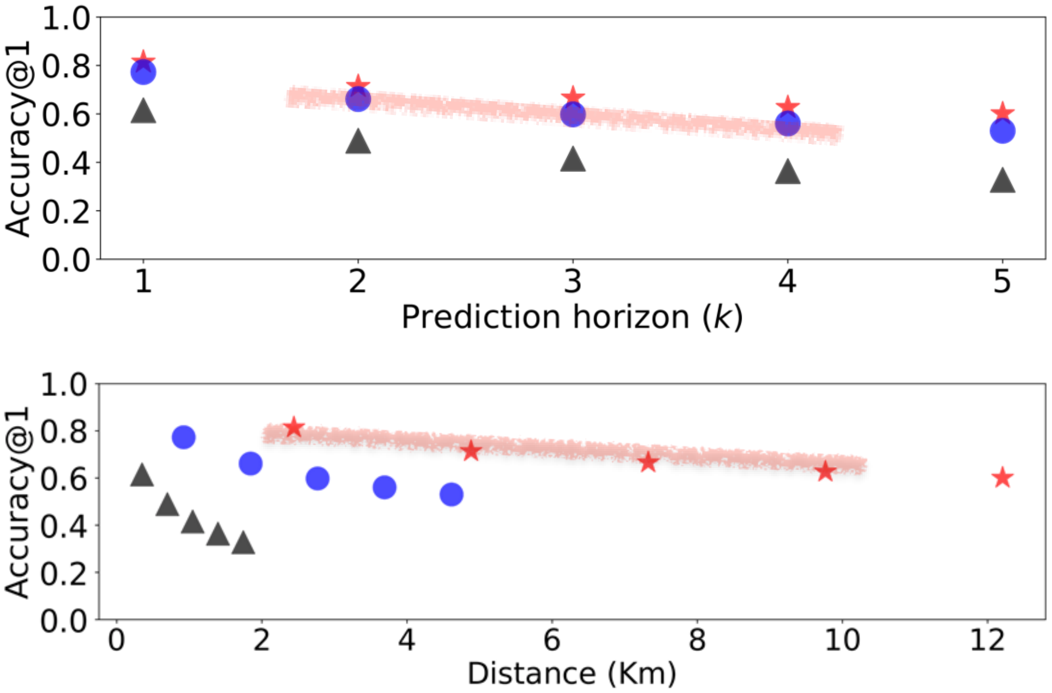
RQ 3) Beam Search Analysis





YORK U

RQ 4) Map Resolution Analysis



7: star

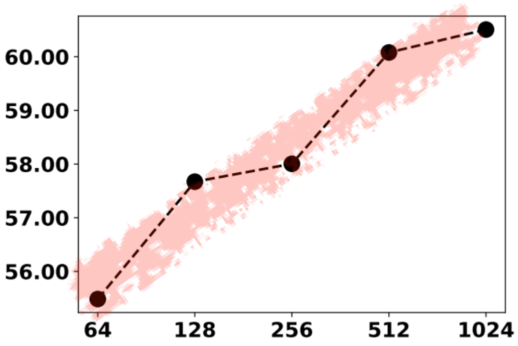
8: circle

9: triangle

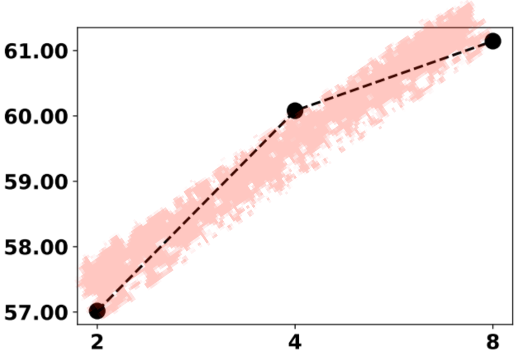
YORK U

RQ 5) Ablation Study

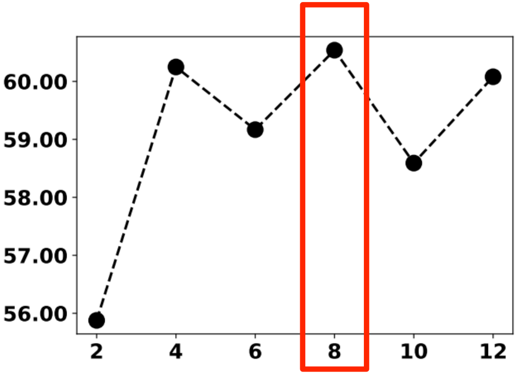
Resolution	Accuracy@1	Accuracy@1 W/O Beam	Change (%)
7	0.4507	0.4365	–3.15
8	0.4244	0.4064	–4.24
9	0.4785	0.4677	–2.26



Hidden dimension



Attention Heads

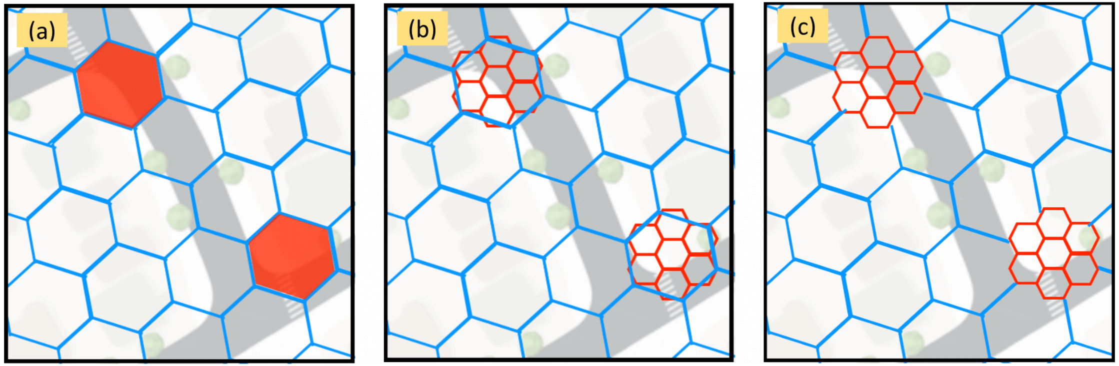


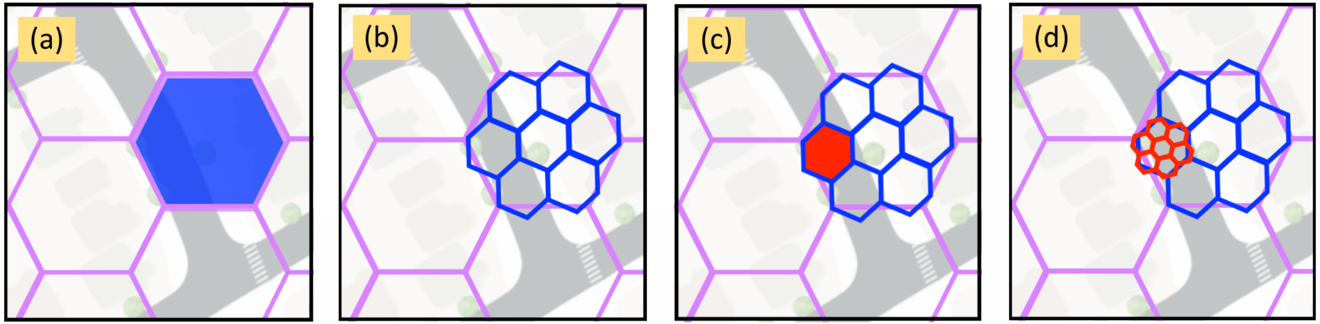
Decoder Blocks

YORK U

Enhancement: Hierarchical Maps

Hierarchical Maps



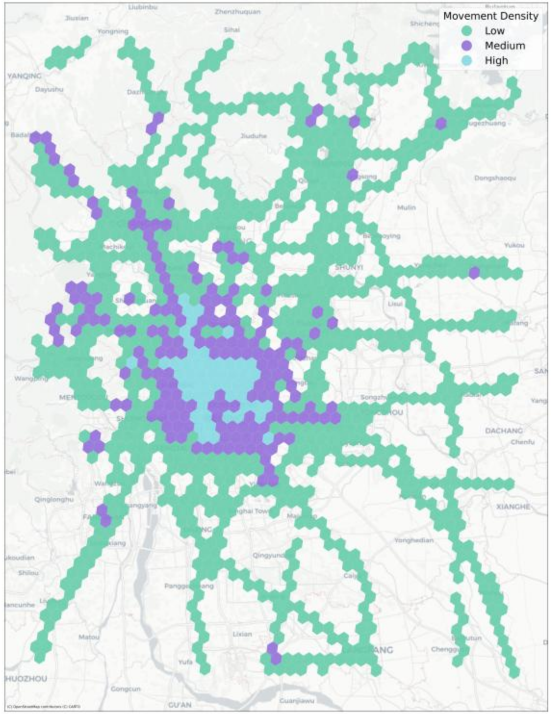


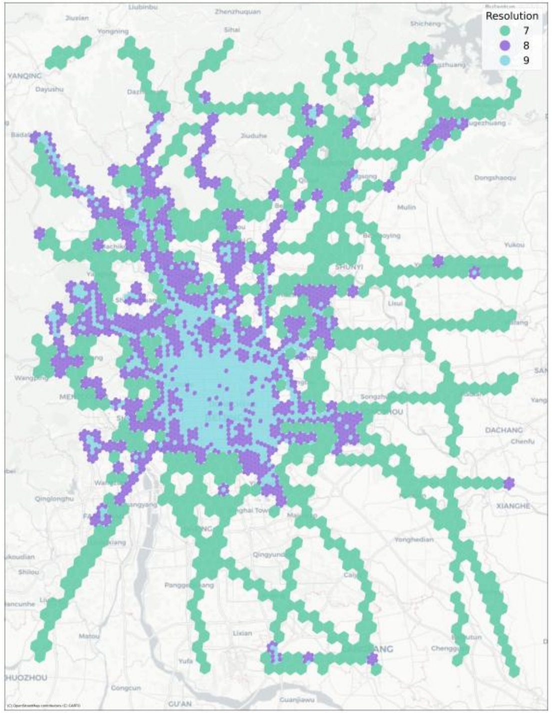
Hierarchical Map Generation Algorithm

Input: Map M, Minimum resolution $R_{\min}$, Maximum resolution $R_{\max}$, Maximum iterations max_iter
Output: Hierarchical map H

```
// Initialize hexagon set at the minimum resolution
H ← Tessellate(M, R_min);
for i ← 1 to max_iter do
    if termination_condition_fn(H) then
        break
    end
    foreach hexagon h ∈ H do
        if splitting_condition_fn(h) and Resolution(h) < R_max then
            Split h into smaller hexagons at the next resolution level;
        end
    end
    Update H with the newly created hexagons;
end
```

Hierarchical Maps

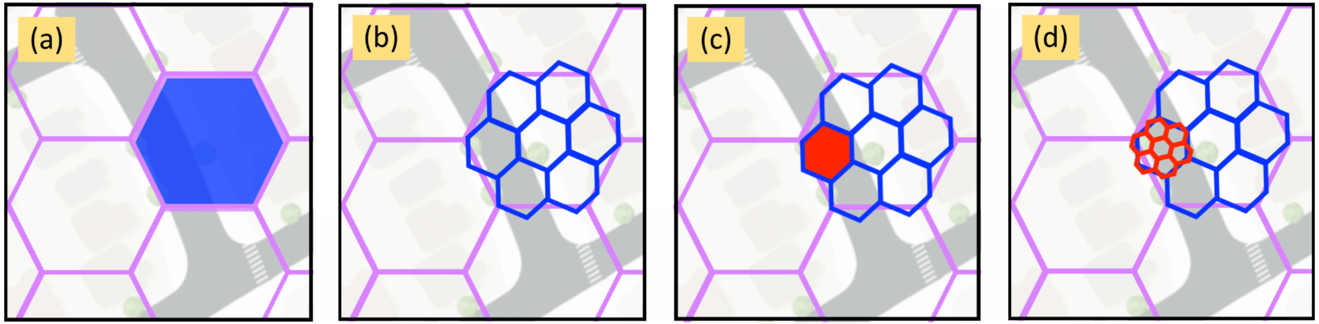




Performance of Trajectory Prediction Using Hierarchical Maps

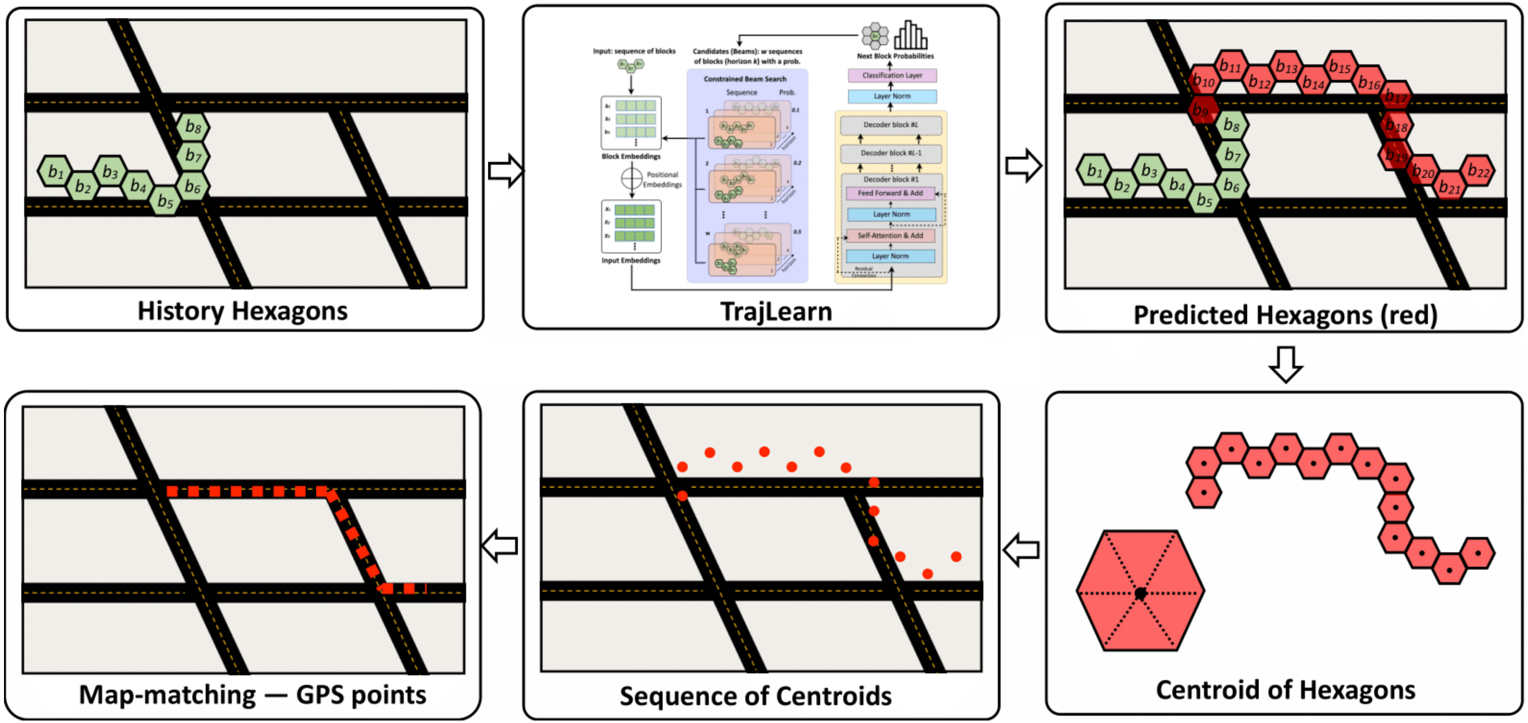
Dataset	Acc@1	Acc@3	Acc@5	BLEU
Ho-Porto	0.4476	0.5333	0.6099	0.5289
Ho-Rome	0.3213	0.4915	0.5605	0.3747
Ho-GeoLife	0.4854	0.5727	0.6496	0.5267

Resolution 7 to 9

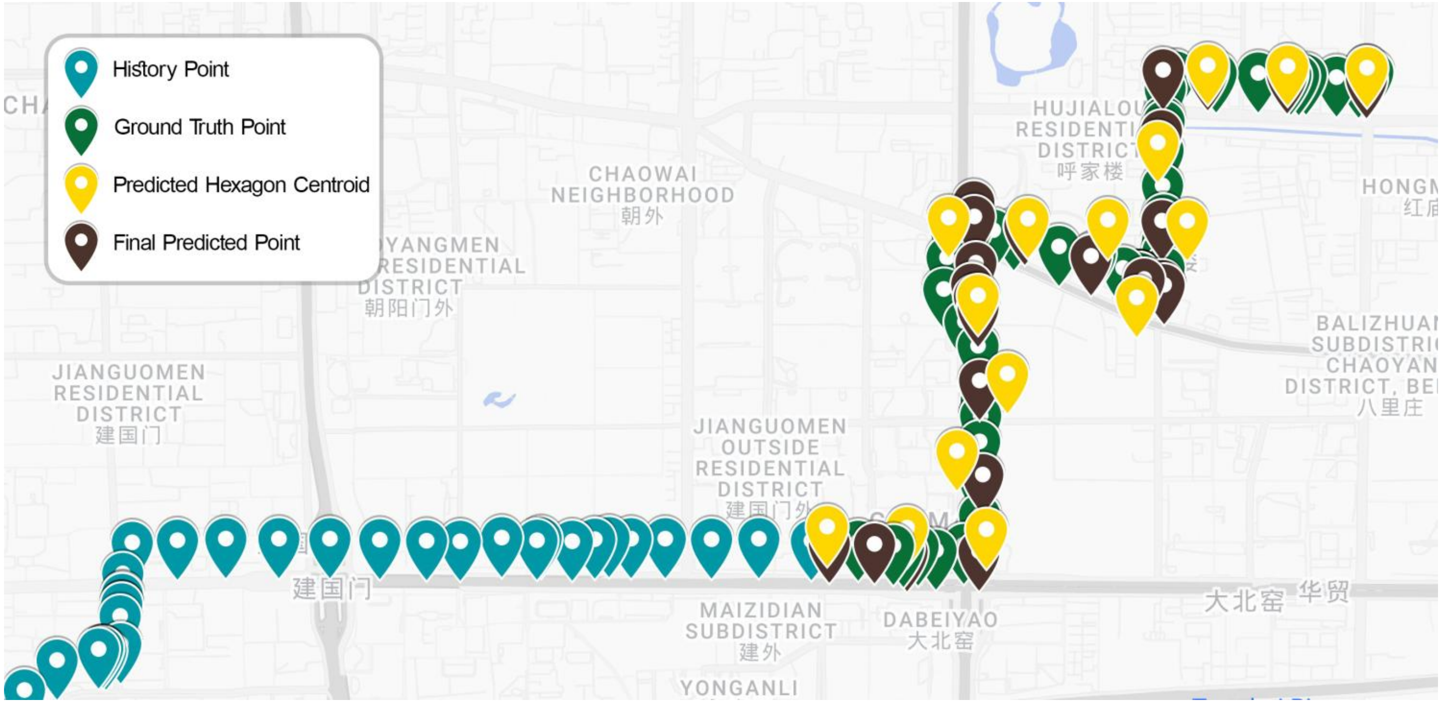


Enhancement: Mapping Predicted Hexagons to GPS Points

Mapping Predicted Hexagons to GPS Points



Mapping Predicted Hexagons to GPS Points: An Example



YORK U

Problem 2: Mobility Network Prediction

Mobility Network Prediction (Revisited)

Let

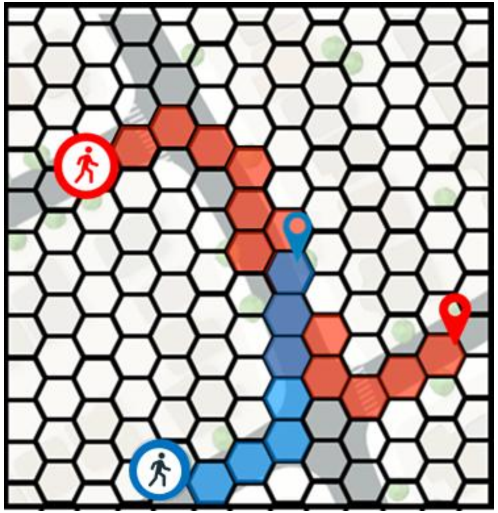
- an observation area $\mathcal{M}$
- a set of objects $\mathcal{N}$ and their history trajectories $\mathcal{T}^l$
- an observation period $[0, \mathcal{W}]$

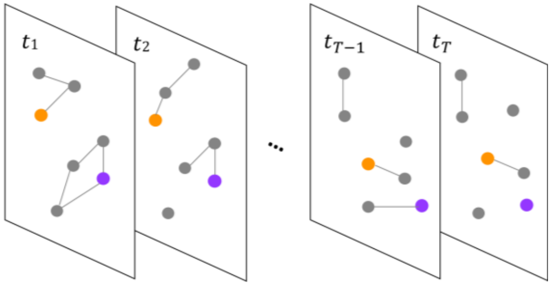
Input: Given

- a prediction horizon $k > 0$

Output: We want to

predict the next k future proximity networks $< G_{i_{w+1}}, G_{i_{w+2}}, \ldots, G_{i_{w+k}} >$





Existing works

Literature Overview

Dynamic Link Prediction

Continuous-Time Dynamic Networks

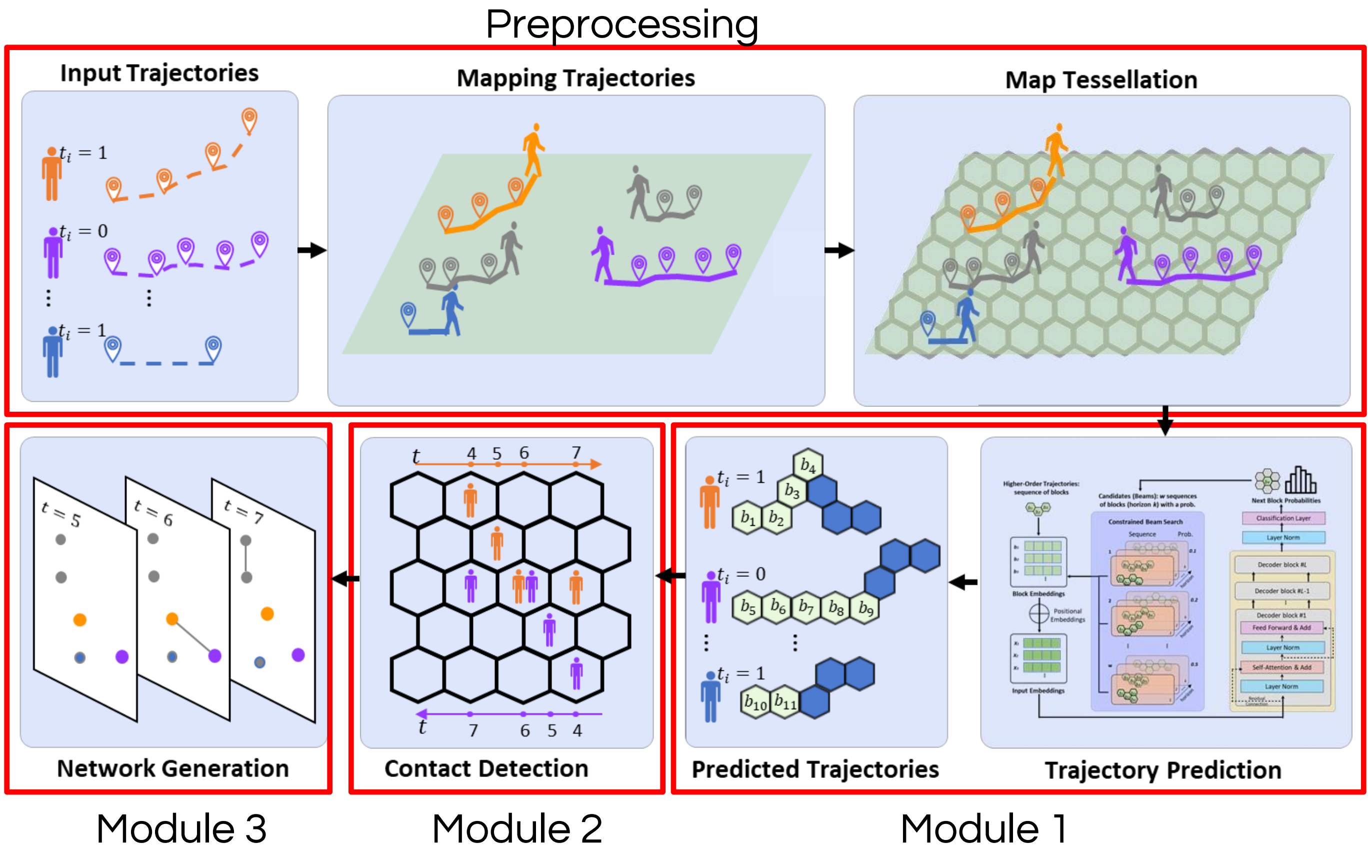
- TGAT
- DyRep
- CAW

Discrete-Time Dynamic Networks

- EvolveGCN
- Roland
- AGCRN
- DyTed

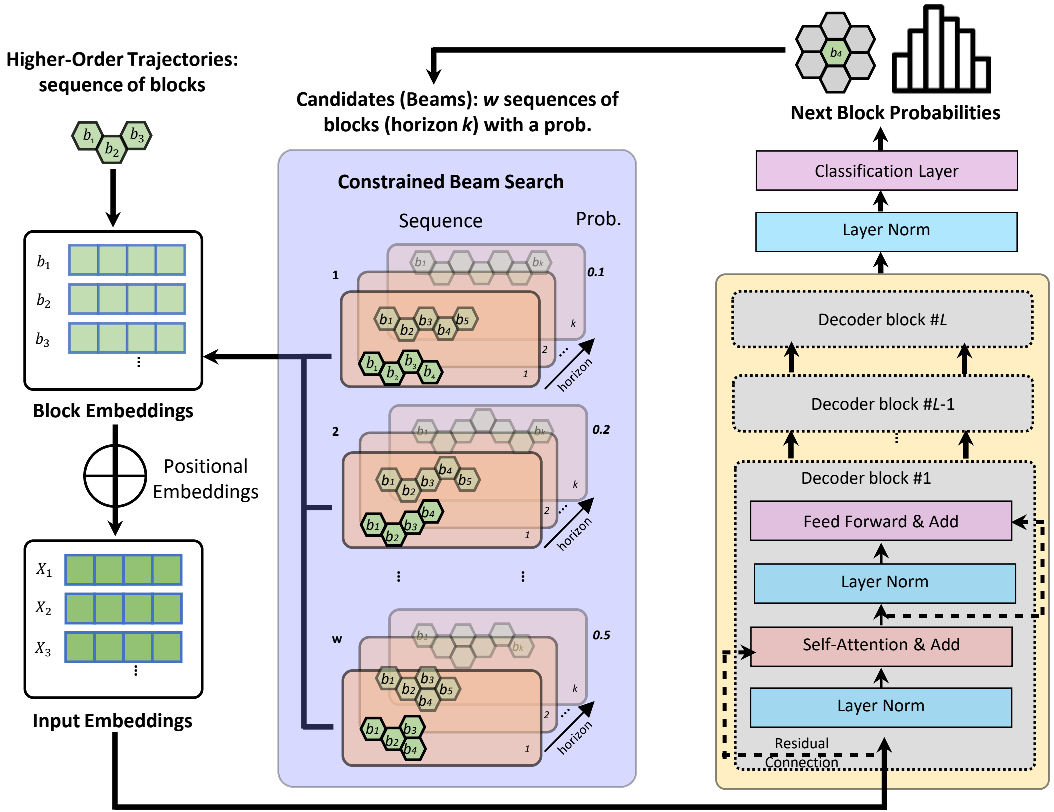
Methodology

MobiNetForecast Overview



YORK U

Trajectory Prediction Module

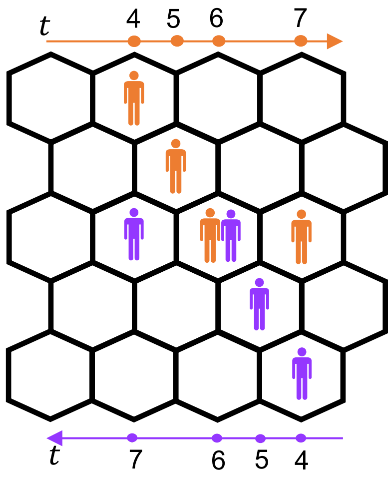


Contact Detection Module

The set of all contacts C_t at a given time step t is collected as:

$$C_{uv}(t) = \begin{cases} 1 & \text{if } b_{u_t} = b_{v_t} \\ 0 & \text{otherwise} \end{cases}$$

$$C_t = \{(u, v) | C_{uv}(t) = 1 \text{ and } u \neq v\}$$



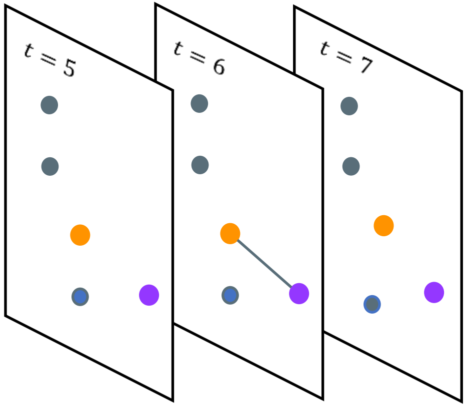
YORK U

Mobility Network Construction Module

For each time step t, we create a network snapshot $G_t = (N_t, E_t)$

$$N_t = \{\, u \mid u \text{ appears in } T^u \text{ at time } t\}$$

$$E_t = \{\, (u, v) \mid (u, v) \in C_t \,\}$$



Assemble a dynamic mobility network G that evolves over time:

$$G = \{\, G_{T+1}, G_{T+2}, \dots, G_{T+k}\}$$

Evaluation

Experimental Scenarios

RQ 1) Framework Performance

- How does MobiNetForecast compare against established baseline methods in predicting dynamic networks?

RQ 2) Model Performance

- How does TrajLearn used in MobiNetForecast compare against using other trajectory prediction methods in predicting dynamic networks?

RQ 3) Map Resolution Analysis

- How does varying the resolution of map tessellation (the size of hexagons) impact the accuracy of contact detection and network construction?

YORK U

Experimental Setup

Computational Environment

- NVIDIA RTX A6000 graphics card and 320GB of memory
- Implementation: Python 3, PyTorch 2.3

Map Tessellation and Resolutions

- H3 geo-indexing resolutions 8 to 10

Training Parameters

- AdamW optimizer with learning rate = 10^{-4} to 5×10^{-9},
- Batch size = 256
- Dropout rate = 0

Datasets

Dataset	Duration	# Objects	# Trajectories	# Contacts@10
GeoLife	4 years	182	17,621	124K
SFCO-3K	1 month	3,000	90,000	1.4M

Timely ordered trajectory data set is split into:

67.5% Training, 7.5% Validation, 25% Testing

Baselines

Temporal Link Prediction-Based Models (RQ1)

- EvolveGCN [Pareja et al., AAAI'20]
- ROLAND [You et al., KDD'22]
- DyTed [Zhang et al., KDD'23]
- AGCRN [Bai et al., NeurIPS'20]

Trajectory Prediction-Based Models (RQ2)

- MC [Gambs et al., MPM'12]
- LSTM
- GRU
- LSTM-ATTN [Luong et al., EMNLP'15]
- Flashback++ [Deng et al., ACM TIST'23]

Our Method

- MobiNetForecast

Metrics

Recall [↑]

- Measures the proportion of actual contacts (true positives) correctly predicted.

$$Recall = \frac{TP}{TP + FN}$$

Precision [↑]

- Quantifies the proportion of correctly predicted contacts among all predicted contacts.

$$Precision = \frac{TP}{TP + FP}$$

F1-Score [↑]

- The harmonic mean of precision and recall, providing a balanced measure of the model's accuracy

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

RQ 1) Framework Performance

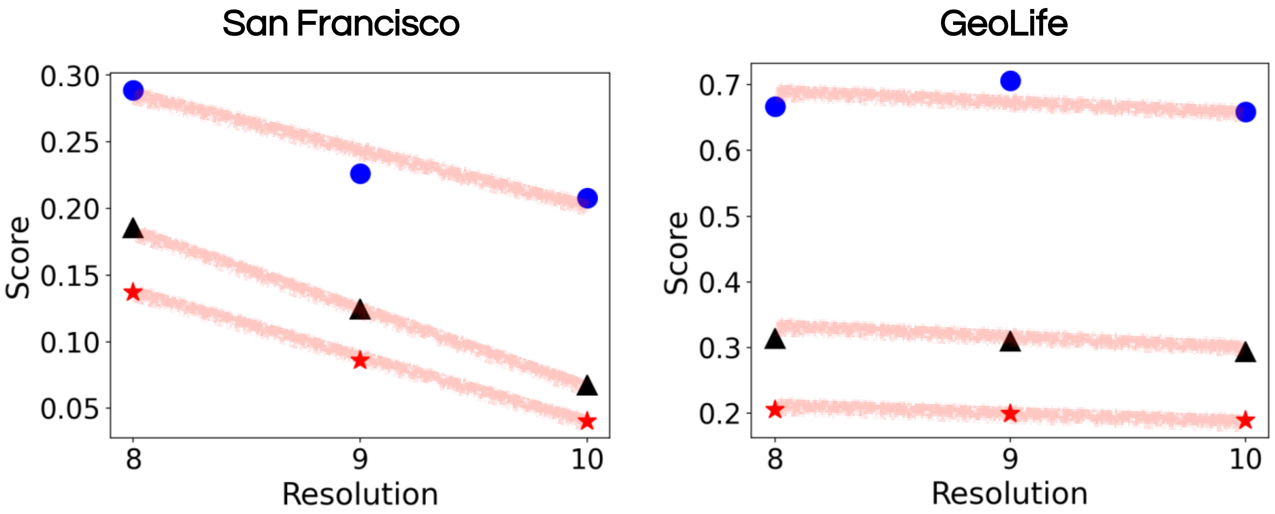
Dataset	Model	Resolution 8			Resolution 9			Resolution 10		
		Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
San Francisco	EvolveGCN	0.0002	0.247	0.0003	0.0002	0.187	0.0003	0.0001	0.119	0.0002
	Roland	0.0006	<u>0.277</u>	0.001	0.0006	0.15	0.001	0.0004	0.127	0.0008
	DyTed	<u>0.001</u>	0.271	<u>0.002</u>	<u>0.001</u>	<u>0.22</u>	<u>0.002</u>	<u>0.0008</u>	<u>0.181</u>	<u>0.002</u>
	AGCRN	0.0006	0.193	0.001	0.0005	0.14	0.0009	0.0004	0.13	0.0008
	MobiNetForecast (ours)	**0.137**	**0.289**	**0.186**	**0.086**	**0.226**	**0.125**	**0.04**	**0.208**	**0.068**
	Improvement	**163×**	**4.21%**	**111×**	**80×**	**4.74%**	**58×**	**52×**	**15.16%**	**44×**
GeoLife	EvolveGCN	0.039	0.207	0.066	0.054	0.214	0.086	0.068	0.221	0.104
	Roland	0.084	0.456	0.143	0.127	0.47	0.201	0.135	0.48	0.21
	DyTed	0.114	<u>0.513</u>	0.187	<u>0.163</u>	<u>0.528</u>	<u>0.25</u>	<u>0.154</u>	<u>0.518</u>	<u>0.238</u>
	AGCRN	<u>0.125</u>	0.437	<u>0.194</u>	0.121	0.455	0.191	0.125	0.479	0.199
	MobiNetForecast (ours)	**0.206**	**0.667**	**0.315**	**0.199**	**0.706**	**0.311**	**0.189**	**0.659**	**0.294**
	Improvement	**65.10%**	**30.05%**	**62.18%**	**21.76%**	**33.84%**	**24.41%**	**22.54%**	**27.13%**	**23.57%**

YORK U

RQ 2) Model Performance

Dataset	Model	Resolution 8			Resolution 9			Resolution 10		
		Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
San Francisco	**MC**	0.023	0.046	0.031	0.015	0.083	0.025	0.008	0.063	0.014
	LSTM	0.097	0.231	0.136	0.065	0.187	0.096	0.026	0.146	0.044
	LSTM-ATTN	0.103	0.245	0.145	0.072	0.193	0.105	0.03	0.158	0.044
	GRU	0.118	0.24	0.158	0.075	0.193	0.108	0.032	0.162	0.053
	Flashback++	0.115	0.261	0.16	0.073	0.2	0.1	0.035	0.187	0.059
	MobiNetForecast (ours)	**0.137**	**0.289**	**0.186**	**0.086**	**0.226**	**0.125**	**0.04**	**0.208**	**0.068**
	Improvement	**16.01%**	**10.88%**	**16.41%**	**14.50%**	**17.02%**	**15.32%**	**15.14%**	**11.28%**	**14.52%**
GeoLife	**MC**	0.013	0.049	0.02	0.01	0.051	0.023	0.008	0.047	0.013
	LSTM	0.143	0.482	0.221	0.132	0.514	0.21	0.137	0.506	0.216
	LSTM-ATTN	0.151	0.546	0.237	0.143	0.531	0.225	0.141	0.523	0.222
	GRU	0.157	0.51	0.24	0.153	0.549	0.239	0.141	0.572	0.227
	Flashback++	0.171	0.553	0.261	0.165	0.586	0.258	0.16	0.535	0.25
	MobiNetForecast (ours)	**0.206**	**0.667**	**0.315**	**0.199**	**0.706**	**0.311**	**0.189**	**0.659**	**0.294**
	Improvement	**20.35%**	**20.60%**	**20.41%**	**20.46%**	**20.62%**	**20.50%**	**16.09%**	**15.18%**	**17.66%**

RQ 3) Map Resolution Analysis



7: Precision

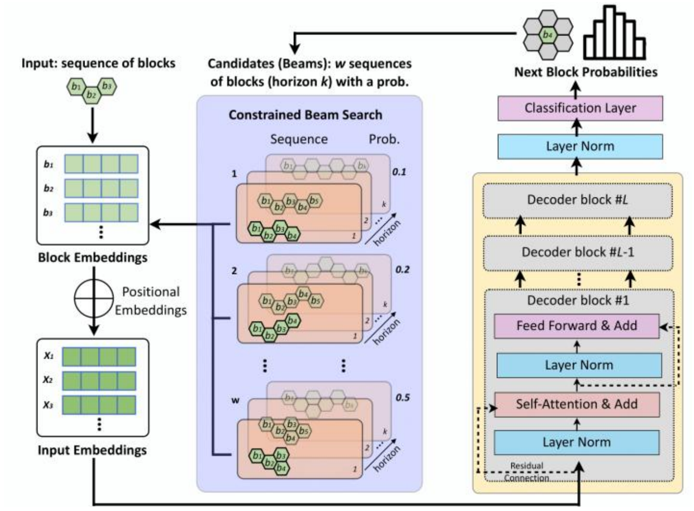
8: Recall

9: F1-Score

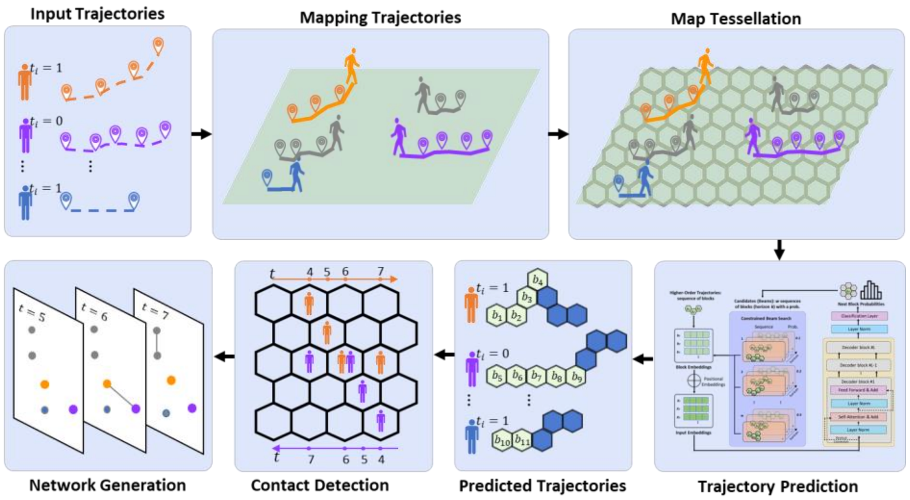
YORK U

Conclusions

Thesis Contributions



TrajLearn



MobiNetForecast

Published work

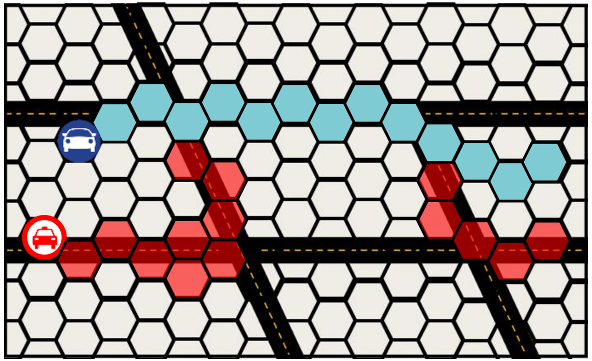
- (Submitted) A. Nadiri, J. Li, G. Abuoda, and M. Papagelis, “Mobility Network Forecasting: A Trajectory-based Contact Prediction Approach”
- (ACM TSAS, 2025) A. Nadiri, J. Li, A. Faraji, G. Abuoda, and M. Papagelis, “TrajLearn: Trajectory Prediction Learning using Deep Generative Models”
- (ACM SIGSPATIAL, 2023) A. Faraji*, J. Li*, G. Alix, M. Alsaeed, N. Yanin, A. Nadiri, and M. Papagelis, “Point2Hex: Higher-order Mobility Flow Data and Resources”

Limitations

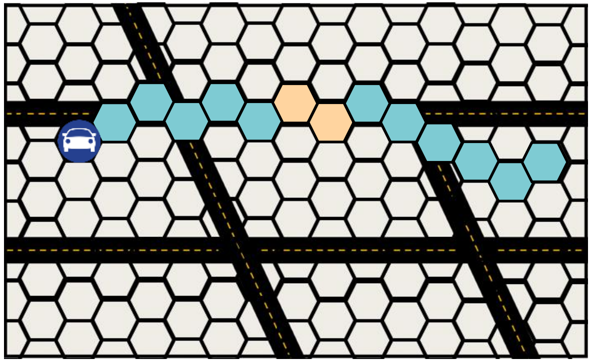
- GPS Data Quality
- Dataset Biases and Coverage
- Data Sensitivity
- Model Interpretability
- Temporal Generalization & Concept Drift

Future Work

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



Trajectory similarity



Trajectory imputation

Thank you!

Questions?