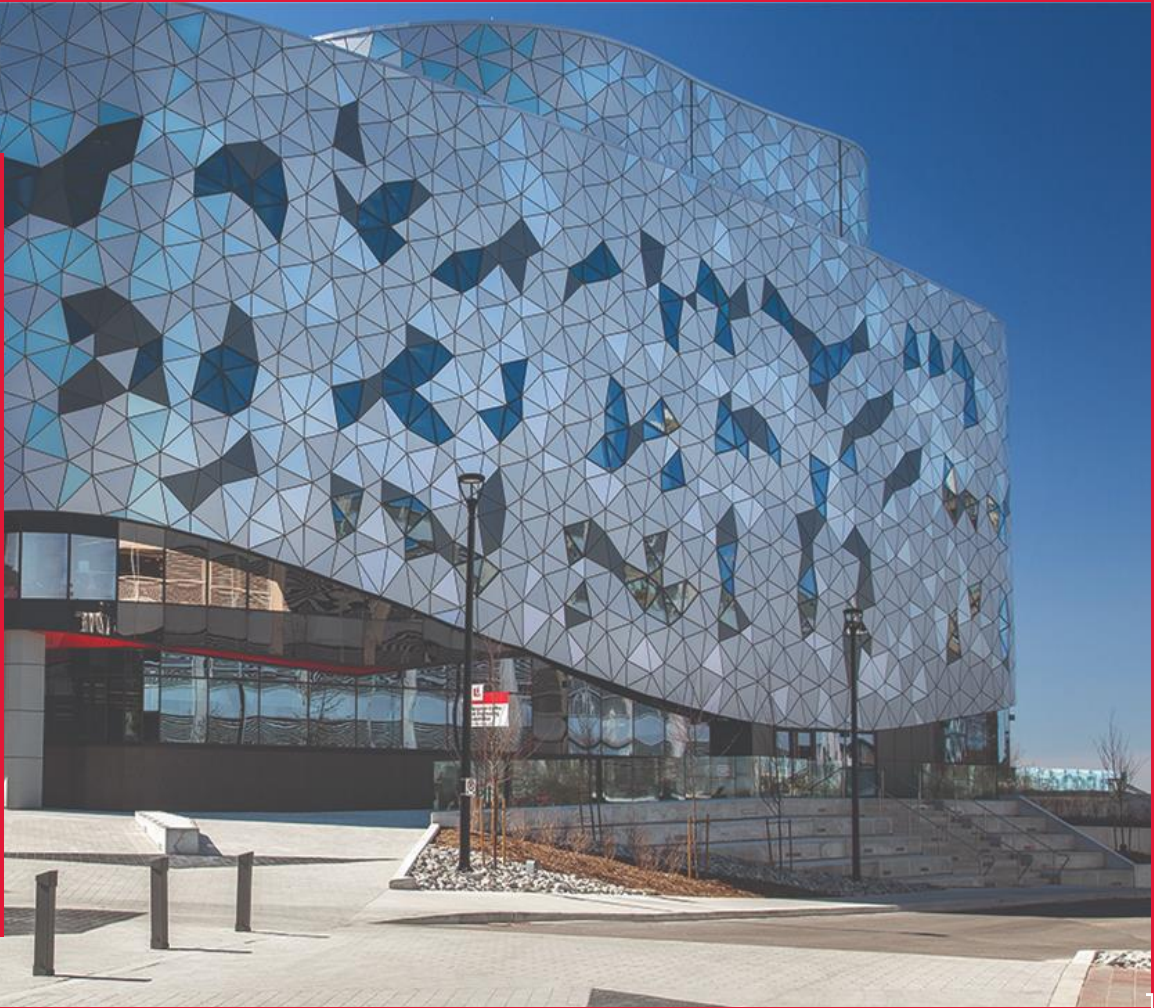
Trajectory Data Mining in the Age of Big Data and AI

Missouri S&T
CS Seminars and Colloquia

Manos Papagelis

Mon, Oct 23, 2023

YORK U



Background & Motivation

Trajectories

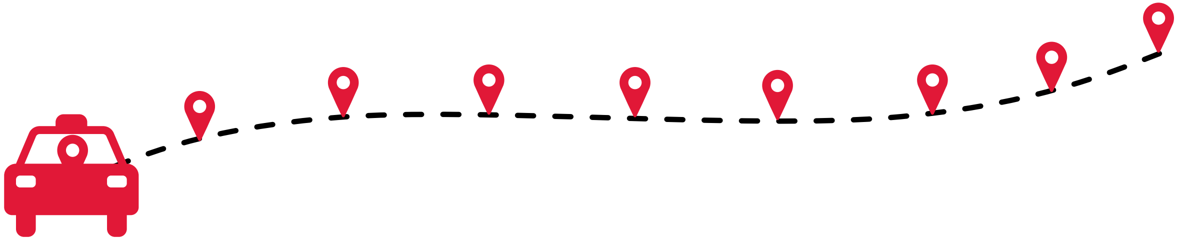
- Trajectory
  - Denoted by τ
  - Represented as:

$$\tau = \langle (x_1, y_1, t_1), \dots, (x_{|\tau|}, y_{|\tau|}, t_{|\tau|}) \rangle$$

object's geo-location

specific time instance

- Trajectory set
  - Consists of all trajectories of all objects
  - Denoted by $\mathcal{T}$



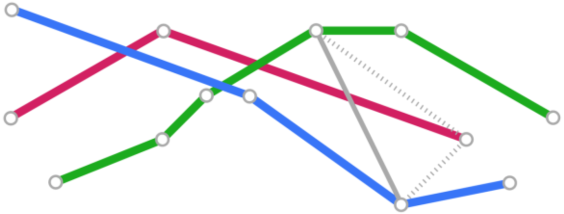
Trajectory Data (or Mobility Data)

- **Massive** trajectory datasets are collected (spatiotemporal data of moving objects)
- Due to **advancement of geolocation tracking** devices
- Motivates various **trajectory analytics**

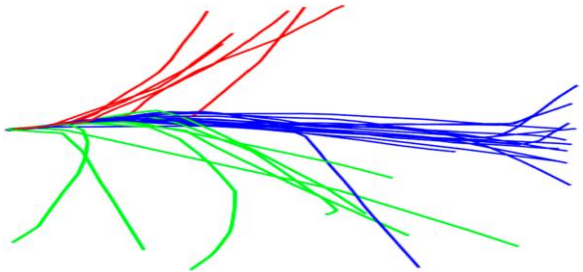
Trajectories contained within the 5th Ring Road in Beijing

Image source: https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/

Trajectory Data Mining



trajectory similarity



trajectory clustering

trajectory anomaly detection
trajectory network mining
trajectory classification
...

challenging computational problems

Trajectory Data Mining in the Age of Big Data and AI

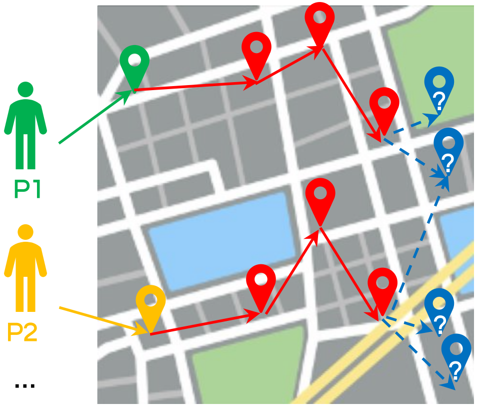
a symbiotic relationship that presents a new strategy for addressing complex problems in trajectory data mining

Image source: This image was created with the assistance of DALL·E 3

YORK U

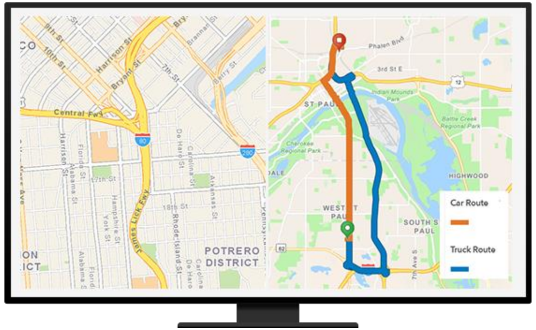
Plethora of Applications

ridesharing



trip/POI (point-of-interest) recommendation

traffic analysis



route planning and optimization

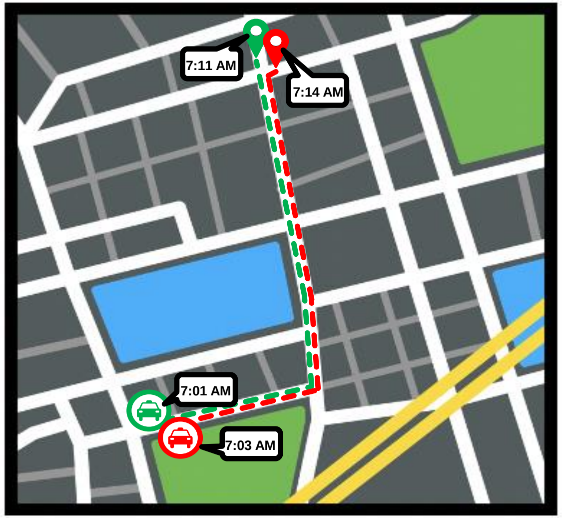
Our Lab’s Journey on Trajectory Data Mining

- Trajectory dataset and resources [ACM SIGSPATIAL ’23]
- Trajectory simplification [ACM SIGSPATIAL ’23]
- Trajectory classification [IEEE MDM ’23]
- Trajectory network analysis [Big Data Research, IEEE MDM '20, GeoInformatica, IEEE BigData '18, 2 x IEEE MDM ’18]
- Mobility + epidemics [ACM SIGSPATIAL/SpatialEpi '24, ACM SIGSPATIAL/SpatialEpi '23, IEEE MDM ’22]
- Transportation optimization [ACM SIGSPATIAL '22, ACM SIGSPATIAL ’22]
- Trajectory prediction [Submitted]
- Trajectory similarity [Submitted]

YORK U

Today's Focus

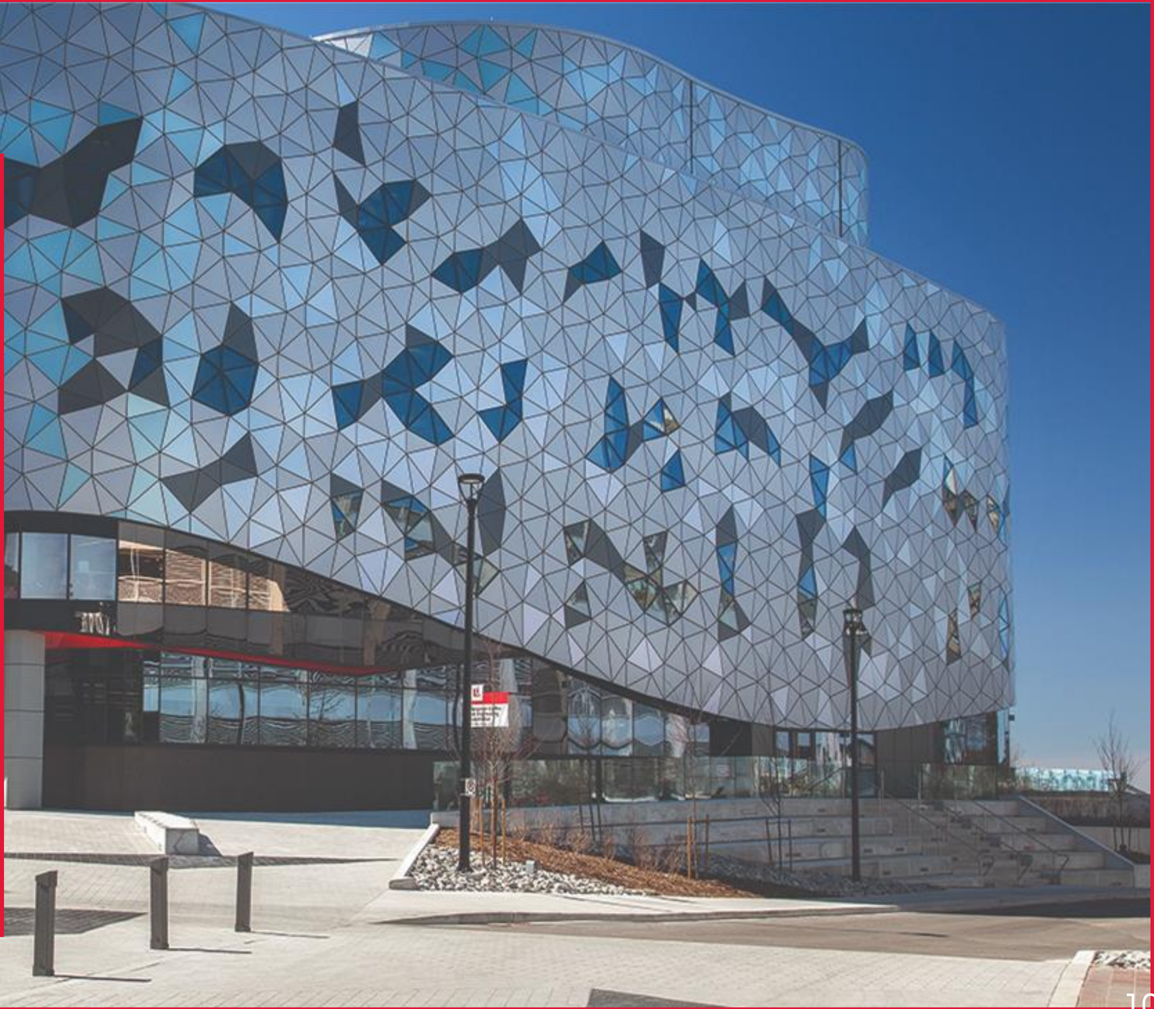
Trajectory Pathlet Dictionary Construction
(Trajectory Simplification)



Trajectory-User Linking
(Trajectory Classification)

Trajectory Pathlet Dictionary Construction

YORK U



Trajectories

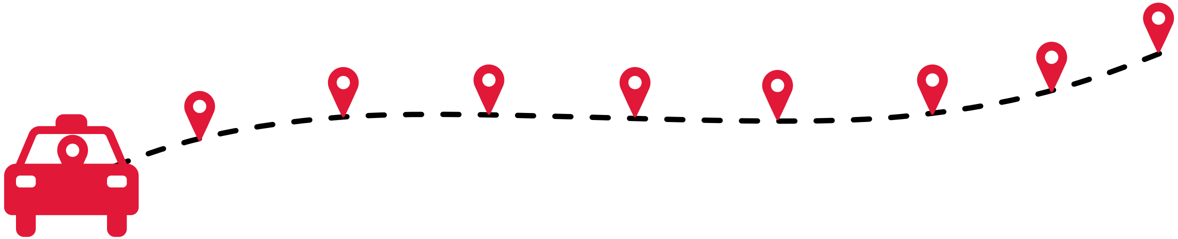
- Trajectory
  - Denoted by τ
  - Represented as:

$$\tau = \langle (x_1, y_1, t_1), \ldots, (x_{|\tau|}, y_{|\tau|}, t_{|\tau|}) \rangle$$

object's geo-location

specific time instance

- Trajectory set
  - Consists of all trajectories of all objects
  - Denoted by $\mathcal{T}$



Trajectories on the Road Network

- Road Segment †
  - Connects two road intersections/ends
  - Denoted by r
  - Collection of all segments $\boldsymbol{R}$

- Modelled as a graph $\mathcal{G}\langle\mathcal{V},\mathcal{E}\rangle$
  - $\mathcal{V}$: **Nodes** (set of road intersections)
  - $\mathcal{E}$: **Edges** (set of road segments)

    $[\mathcal{E} = \boldsymbol{R} \subseteq \mathcal{V} \times \mathcal{V}]$

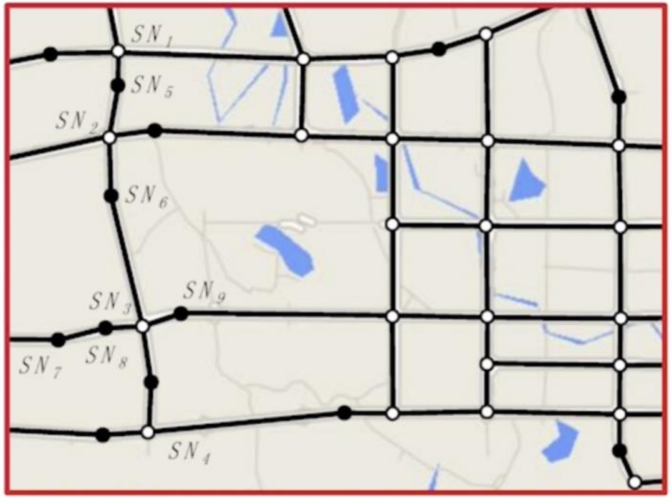
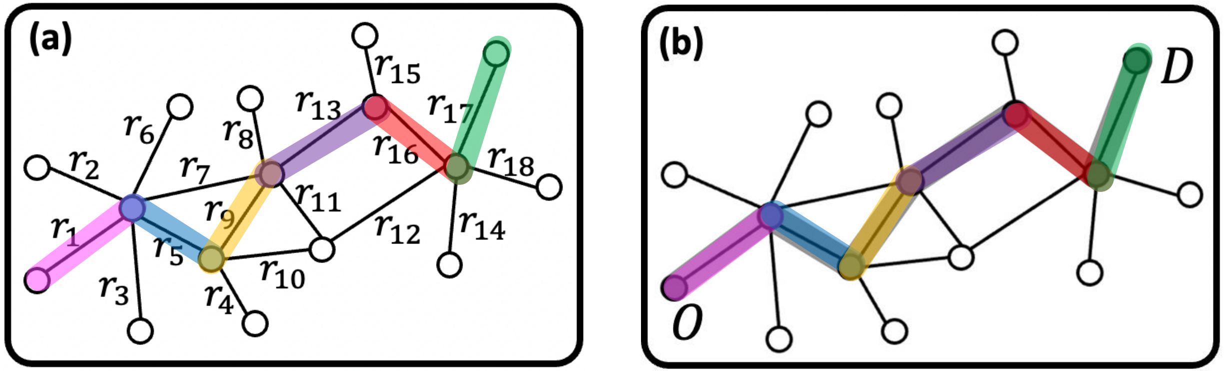


Image Source: "Updating Road Networks by Local Renewal from GPS Trajectories" [Wu et al, MDPI '16]

† "Road", "segment" and "road segment" are terms used interchangeably.

Road Segment-based Representation

- Each trajectory τ can be expressed as a set of road segments $\boldsymbol{R}_s \subseteq \boldsymbol{R}$
- This special representation is denoted by $\mathfrak{R}(\tau)$



$$\mathfrak{R}(\tau) = \{r_1, r_5, r_9, r_{13}, r_{16}, r_{17}\}$$

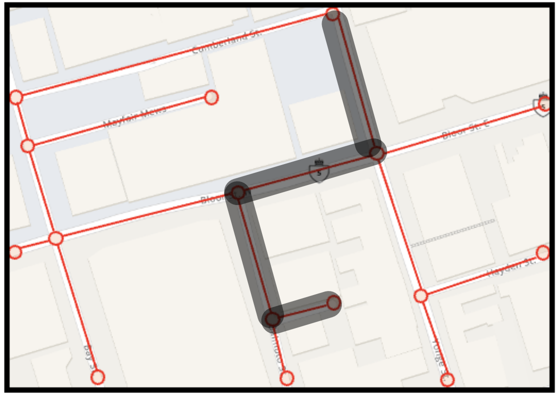
YORK U

Trajectory Pathlet Dictionary (PD) Construction

- Constructing a small set of basic building blocks that can represent a wide range of trajectories
- Many names in the literature

  [Panagiotakis et al – TKDE '12, Chen et al – SIGSPATIAL '13, Sankararaman et al – SIGSPATIAL '13, Agarwal et al – PODS '18, Li et al – TSAS '18, Zhao et al – CIKM '18]

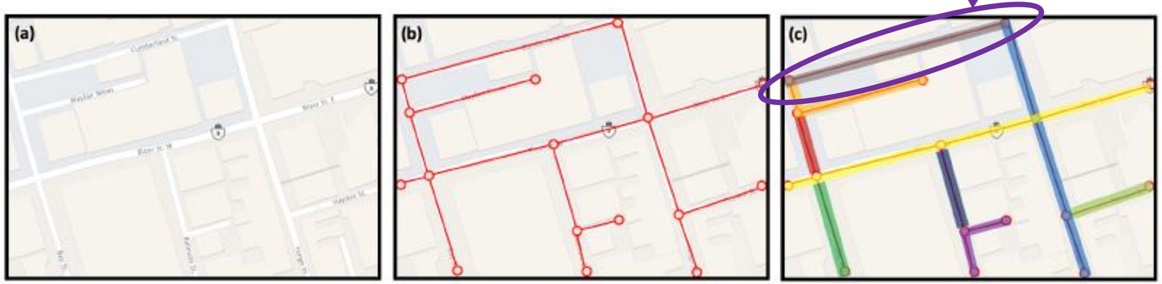
  - Pathlet
  - Subtrajectory
  - Trajectory Segments
  - Fragments
  - ...



YORK U

Brief Background: Pathlets

Grey pathlet has two neighbors: orange and blue pathlets

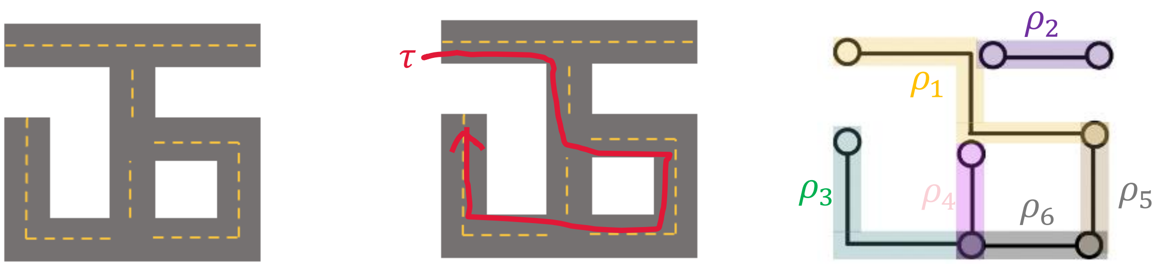


- Pathlet (ρ) - any sub-path in the road network $\mathcal{G}$
  - Collection of all pathlets $\mathcal{P}$ (a pathlet set)
  - Edge-disjoint – no two pathlets overlap in edges
- Pathlet Length
  - Denoted by ℓ; the path length in the road network ($\ell \geq 1, \ell \in \mathbb{Z}$)
  - χ-order Pathlet Set – All pathlets have length at most χ
- Pathlet Graph – derived from the road network $\mathcal{G}$, denoted by $\mathcal{G}_p\langle \mathcal{V}_p, \mathcal{E}_p \rangle$
- Pathlet Neighbors – share the same start/end points (road intersections)
  - Neighbor set - denoted by $\Psi(\rho)$; the collection of all neighbors of ρ

YORK U

Pathlet-based Representation of a Trajectory

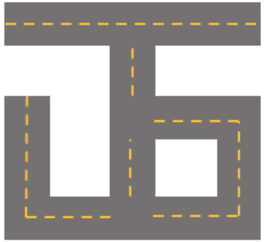
Denoted by $\Phi(\tau) = \{\rho^{(1)}, \rho^{(2)}, \ldots, \rho^{(k)}\}$



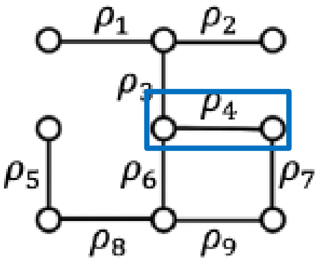
(a) (b) (c)

$\Phi(\tau) = \{\rho_1, \rho_5, \rho_6, \rho_3\}$

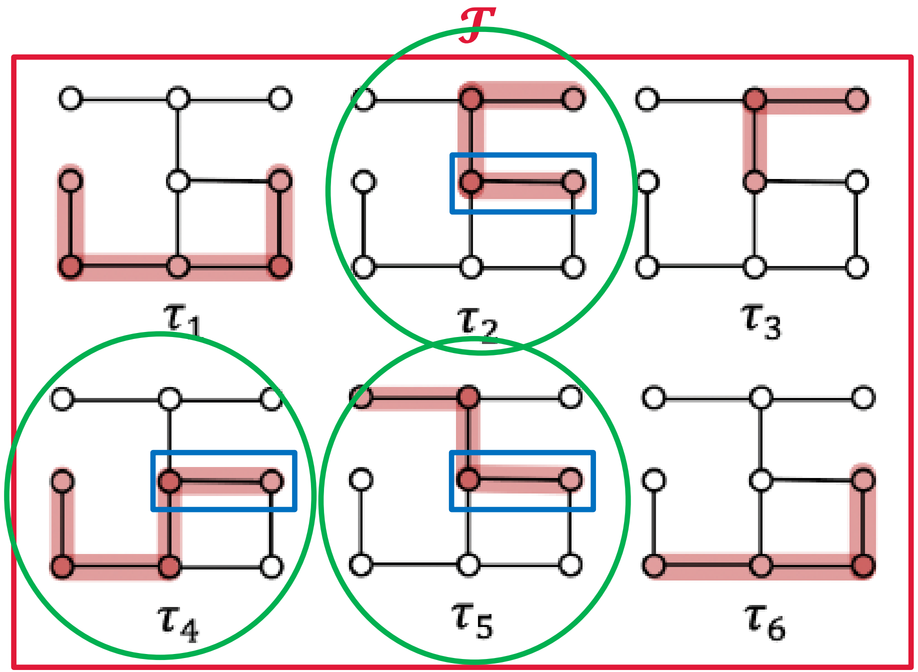
Trajectory Traversal Set



(a)



(b)



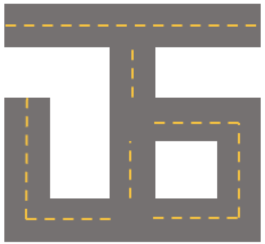
- Denoted by

$$\Lambda(\rho) = \{\tau \mid \forall \tau \in \mathcal{T}, \rho \in \Phi(\tau)\}$$

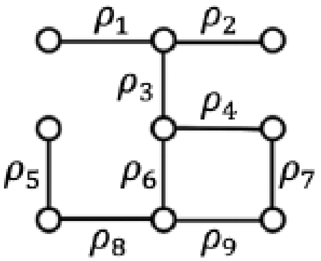
- Pathlet Weights – importance in the road network

$\Lambda(\rho_1) = \{\tau_5\}$	$\Lambda(\rho_4) = \{\tau_2, \tau_4, \tau_5\}$	$\Lambda(\rho_7) = \{\tau_1, \tau_6\}$
$\Lambda(\rho_2) = \{\tau_2, \tau_3\}$	$\Lambda(\rho_5) = \{\tau_1, \tau_4\}$	$\Lambda(\rho_8) = \{\tau_1, \tau_4, \tau_6\}$
$\Lambda(\rho_3) = \{\tau_2, \tau_3, \tau_5\}$	$\Lambda(\rho_6) = \{\tau_4\}$	$\Lambda(\rho_9) = \{\tau_1, \tau_6\}$

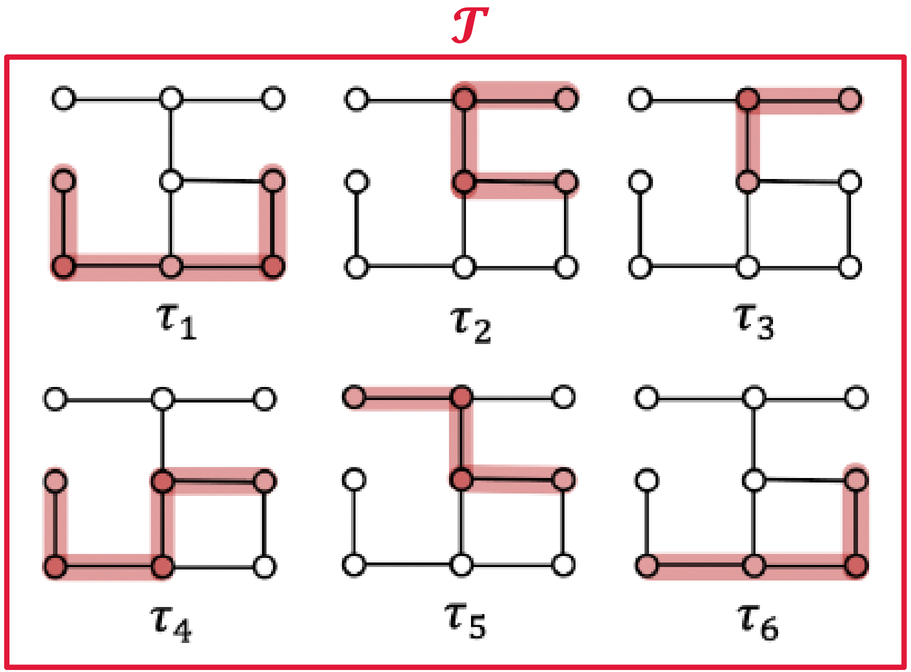
Pathlet Dictionary



(a)



(b)



pathlets
(keys)

ρ_1	$\{\tau_5\}$
ρ_2	$\{\tau_2, \tau_3\}$
ρ_3	$\{\tau_2, \tau_3, \tau_5\}$
ρ_4	$\{\tau_2, \tau_4, \tau_5\}$
ρ_5	$\{\tau_1, \tau_4\}$
ρ_6	$\{\tau_4\}$
ρ_7	$\{\tau_1, \tau_6\}$
ρ_8	$\{\tau_1, \tau_4, \tau_6\}$
ρ_9	$\{\tau_1, \tau_6\}$

trajectory traversal set
(values)

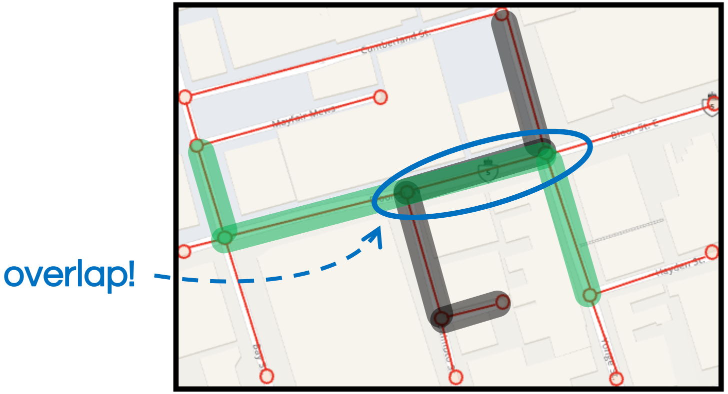
Existing Works

Existing Works and Limitations

- Existing works

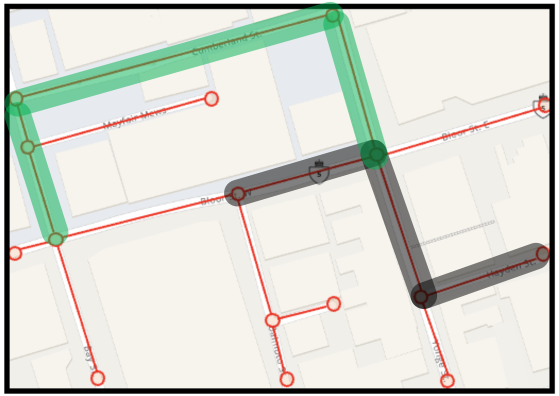
  [Panagiotakis et al – TKDE '12, Chen et al – SIGSPATIAL '13, Sankararaman et al – SIGSPATIAL '13, Agarwal et al – PODS '18, Li et al – TSAS '18, Zhao et al – CIKM '18]

- Main Limitations
  - Traditional-based (non-learning) methods
  - Overlapping pathlet assumption



Overlapping Pathlets

(Top-down Approach)

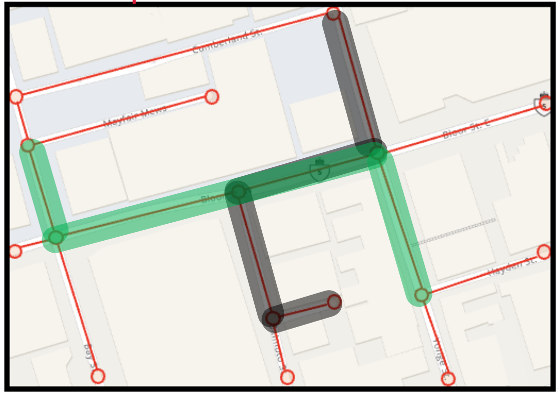


Edge-disjoint Pathlets

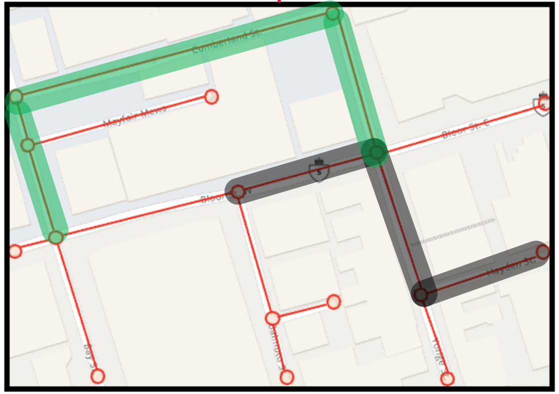
(Bottom-up Approach)

Top-down vs Bottom-up Methods

Top-down Methods



Bottom-up Methods



- Candidates are all pathlets of various sizes and configurations
- Reduce dictionary size by considering only the top most (popular) ones
- Expensive space complexity: $\Theta(n^2)$

- Candidates are all length-1 pathlets (road segments)
- Form the dictionary by merging neighbor (adjacent) pathlets
- Space efficient: $\Theta(n)$

Space complexities can be proven theoretically

the number of road segments

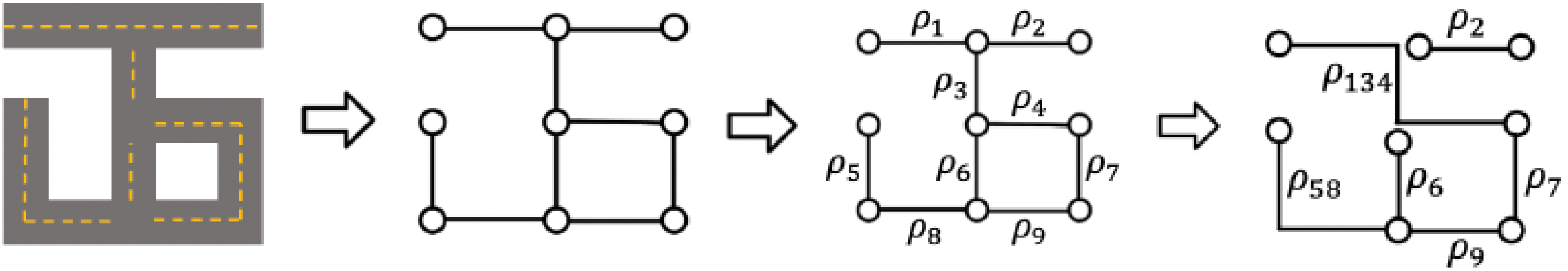
YORK U

Novel Trajectory Metrics

- Trajectory Representability
  - Denoted by $\mu \in [0\%, 100\%]$
  - The percentage of a trajectory that can be represented using pathlets in the pathlet set
  - $\mu(\tau) = \frac{|\Phi(\tau)|}{\ell(\tau)}$
- Trajectory Loss
  - Denoted by L_{traj}
  - The percentage of all trajectories with representability of 0%

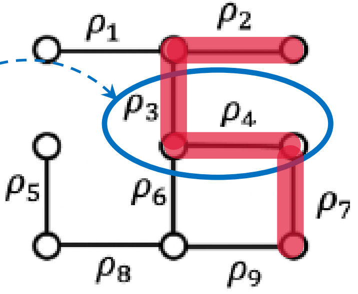
YORK U

Trajectory Representability and Loss - Example



After the merging-based algorithm

Pathlet ρ_3 and ρ_4 are no longer part of $\Phi(\tau)$, since it merged with ρ_{134} and $\{\rho_1\}$ are not in $\Phi(\tau)$



τ

$$\Phi(\tau) = \cancel{\{\rho_2, \rho_3, \rho_4, \rho_7\}} \quad \{\rho_2, \rho_7\}$$

$$\mu(\tau) = \cancel{100\%} \quad 50\%$$

Notice that μ is monotonically non-increasing at each step of the iteration

Trajectory is lost/discarded once μ reaches zero!

YORK U

Pathlet Dictionary Construction - Objectives

Objective	Mathematical Notation	Associated Weight
(O1)	$\min \lvert\mathbb{S}\rvert$	α_1
(O2)	$\min \phi = \min \frac{1}{\lvert\mathcal{T}\rvert} \sum_{\tau \in \mathcal{T}} \lvert\Phi(\tau)\rvert$	α_2
(O3)	$\min L_{traj}$	α_3
(O4)	$\max \bar{\mu} = \max \frac{1}{\lvert\mathcal{T}\rvert} \sum_{\tau \in \mathcal{T}} \mu(\tau)$	α_4

(O1) Minimal size of candidate pathlet set $\mathbb{S}$

(O2) Minimal average number of pathlets representing each trajectory, ϕ

(O3) Minimal trajectory loss

(O4) Maximal average representability values for the remaining trajectories, $\bar{\mu}$

$$\boxed{\min_{\alpha_1+\alpha_2+\alpha_3+\alpha_4=1} \left(\alpha_1 |\mathbb{S}| + \alpha_2 \cdot \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} |\Phi(\tau)| + \alpha_3 L_{traj} - \alpha_4 \cdot \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \mu(\tau) \right)}$$

Problem Statement

Problem Statement

- Trajectory Pathlet Dictionary Construction
  - **Given**: Trajectory set $\mathcal{T}$

    Road Network $\mathcal{G}$ of map $\mathcal{M}$

    Maximum pathlet length $\chi \geq 1$

    Maximum trajectory loss M

    Average trajectory representability threshold $\hat{\mu}$
  - **Construct** a trajectory pathlet dictionary denoted by $\mathbb{S}$
  - Constraints:

    All pathlets in $\mathbb{S}$ are edge-disjoint and have lengths $\ell \leq \chi$

    Achieve the maximum possible utility based on our objective

    Trajectory loss constraint $L_{traj} < M$

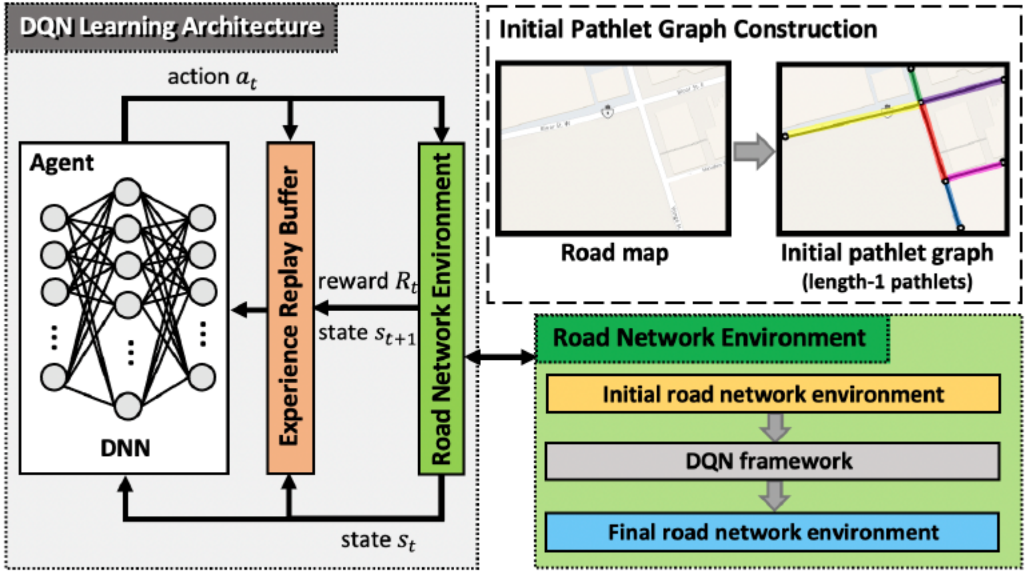
    Trajectory representability constraint $\bar{\mu} \geq \hat{\mu}$

YORK U

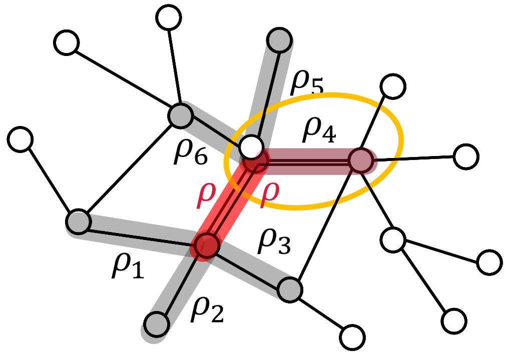
Methodology - PathletRL

PathletRL - Overview

- Extracting candidate pathlets
- Deep Reinforcement Learning framework



Extracting Candidate Pathlets - Example



ρ_{merge}	Utility
MERGED(ρ, ρ_1)	+0.7
MERGED(ρ, ρ_2)	+1.8
MERGED(ρ, ρ_3)	-1.6
MERGED(ρ, ρ_4)	+5.5
MERGED(ρ, ρ_5)	-3.2
MERGED(ρ, ρ_6)	+2.9

Deep Reinforcement Learning

Deep Reinforcement Learning (DRL) Framework and Components

- Desirable actions
  - Lead to higher rewards
- Unfavorable actions
  - Lead to punishment (Lower-valued rewards)
- Idea
  - Learn the best sequence of actions that yield the maximum possible reward value
- Components
  - The Environment and the Agent
  - The States and Actions
  - The Reward Function (Utility)
  - The Reinforcement Learning Policy
  - The Experience Replay Buffer

DRL Components: The Environment and the Agent

- Environment
  - The pathlet graph $\mathcal{G}_p$
  - It is where the algorithm will be operating on
- Agent
  - Our agent is trained to learn which pathlets in the pathlet graph are to be merged/kept unmerged
  - The agent is trained to learn the most optimal sequence of actions that yield the highest possible utility in the form of rewards

DRL Components: The State and Action Spaces

- The State Space $s_t = (S_1, S_2, S_3, S_4) \in \mathcal{S} = \mathbb{R}^4_{\geq 0}$
  - S_1 - the number of pathlets in the current pathlet graph
  - S_2 - the average number of pathlets to represent the trajectories
  - S_3 - the trajectory loss
  - S_4 - the average trajectory representability
- The Action Space
  - $a_t \in \mathcal{A} = \{KEEP, MERGE\}$
  - Merge action requires the agent to merge the current pathlet ρ with one of its $|\Psi(\rho)|$ neighbors
  - Write our action space as:

$$\mathcal{A} = \bigcup_{\forall \hat{\rho} \in \Psi(\rho)} MERGE(\rho, \hat{\rho}) \cup \{KEEP(\rho)\}$$

YORK U

DRL Components: The Reward Function

- The Reward Function

$$\max_{a_t} \mathbb{E}\left[\left(-\alpha_1|\mathbb{S}| - \alpha_2\frac{1}{|\mathcal{T}|}\sum_{\tau\in\mathcal{T}}|\Phi(\tau)| - \alpha_3 L_{traj} + \alpha_4\frac{1}{|\mathcal{T}|}\sum_{\tau\in\mathcal{T}}\mu(\tau)\right)\right] \qquad (*)$$

- Instantaneous Rewards

$$r_t = -\alpha_1\Delta|\mathbb{S}| - \alpha_2\Delta\phi - \alpha_3\Delta L_{traj} + \alpha_4\Delta\bar{\mu}$$

The change in value between the previous and current timesteps

- Discount Rate Factor
  - Realize the importance of both immediate and long-term rewards
  - $\gamma \in [0,1]$

YORK U

DRL Components: The Policy and Deep Q Networks (DQNs)

- Goal: learn the most optimal policy π through the selection of $a_t \in \mathcal{A}$ while in state $s_t \in \mathcal{S}$ that maximizes the Q-index
- Q-learning
  - Agent records and keeps track of all possible (s_t, a_t) pairs and the associated Q-values in a lookup table
  - The Q-table is updated at each timestep recursively:

$$Q^{\pi}(s_t, a_t) \leftarrow Q^{\pi}(s_t, a_t) + \alpha_{lr}\left[\gamma \max_{a_{t+1}} Q^{\pi}(s_{t+1}, a_{t+1}) - Q^{\pi}(s_t, a_t)\right]$$

The learning rate

- Non-linear approximator
  - State-space is continuous
  - Unable to maintain large state-action tables
  - Deep Q Networks!

DRL Components: The Experience Replay Buffer

- Learning based on prior experience

- Collection of data
  - Keeping track of all state-action pairs/state-transitions
  - Learn later

- The experience tuple records (s_t, a_t, r_t, s_{t+1}) are stored in a memory buffer (the experience replay buffer)
  - The agent samples a memory minibatch from this replay buffer

Evaluation - PathletRL

Evaluating PathletRL

RQ 1) Quality of Dictionary

- How does PathletRL compare with SotA methods?

RQ 2) Memory Storage Needs

- How much memory does the bottom-up approach save compared to top-down?

RQ 3) Ablation Study

- How much more effective is PathletRL against its ablation versions?

RQ 4) Partial Trajectory Reconstruction

- How effective is the constructed PD in reconstructing original trajectories?

Datasets

- TORONTO
  - Realistic synthetic car traffic dataset generated using SUMO app[†]
- ROME
  - Real world taxi cab trajectories taken from CRAWDAD[‡]

	TORONTO	ROME
# nodes	~1.9K	~7.5K
# edges/initial pathlets	~2.5K	~15.4K
# trajectories	~169K	~3.8M
Observation period	3.7 hours	1 week

- 70% for training (constructing the PD); 30% for testing (evaluating the PD)

[†] SUMO (Simulation of Urban Mobility): https://www.eclipse.org/sumo/ - an application for simulating traffic
[‡] CRAWDAD: https://crawdad.org/ - an archive site for wireless network and mobile computing datasets

Baselines

- SotA
  - Chen et al. [Chen et al, SIGSPATIAL '13] — Solvable with dynamic programming
  - Agarwal et al. [Agarwal et al, PODS '18] — Framed as subtrajectory clustering problem
- Null Model
  - SGT — Length-1 pathlets only (no merging occurs)
- Ablation Versions
  - PathletRL-RND
  - PathletRL-NR
  - PathletRL-UNW

PathletRL Algorithm	Representability Measure	Weighted Networks	Deep Learning Policy
PathletRL-Nr	✗	✓	✓
PathletRL-Rnd	✓	✓	✗
PathletRL-Unw	✓	✗	✓
PathletRL (ours)	✓	✓	✓

Evaluation Metrics

- $|\mathbb{S}|$, the size of the pathlet dictionary
- ϕ, the average number of pathlets that represent each trajectory
- L_{traj}, the average number of trajectories discarded (%)
- $\bar{\mu}$, the average representability across the remaining trajectories (%)

Notes:

- For the first three metrics lower values are better; for the last one higher values are better
- The third and fourth metrics are not applicable to [Chen et al, SIGSPATIAL '13] and [Agarwal et al, PODS '18]
- The fourth metric is not applicable to PathletRL-NR

YORK U

RQ 1) Numerical Results and Key Observations

		Baselines		Null	PathletRL				% Impr.
		[26]	[1]	SGT	Rnd	Nr	Unw	(ours)	
Toronto	$\lvert\mathbb{S}\rvert$	13,886	7,982	2,563	2,454	1,896	1,801	**1,743**	+3.22%
	ϕ	7.02	5.97	4.76	3.77	**2.89**	3.98	3.75	−22.9%
	L_{traj}	N/A	N/A	0%	19.7%	17.6%	**15.1%**	15.2%	−0.66%
	$\bar{\mu}$	N/A	N/A	100%	79.9%	N/A	80.0%	**83.9%**	+4.88%
Rome	$\lvert\mathbb{S}\rvert$	59,396	31,017	15,465	9,718	7,003	5,804	**5,291**	+8.84%
	ϕ	202.91	188.33	230.15	173.04	158.18	146.39	**139.89**	+4.44%
	L_{traj}	N/A	N/A	0%	24.9%	21.1%	22.9%	**20.4%**	+3.32%
	$\bar{\mu}$	N/A	N/A	100%	82.7%	N/A	**86.2%**	85.6%	−0.70%

- PathletRL improves from the null model, SGT
- PathletRL outperforms traditional methods ([Chen et al, SIGSPATIAL '13] and [Agarwal et al, PODS '18])

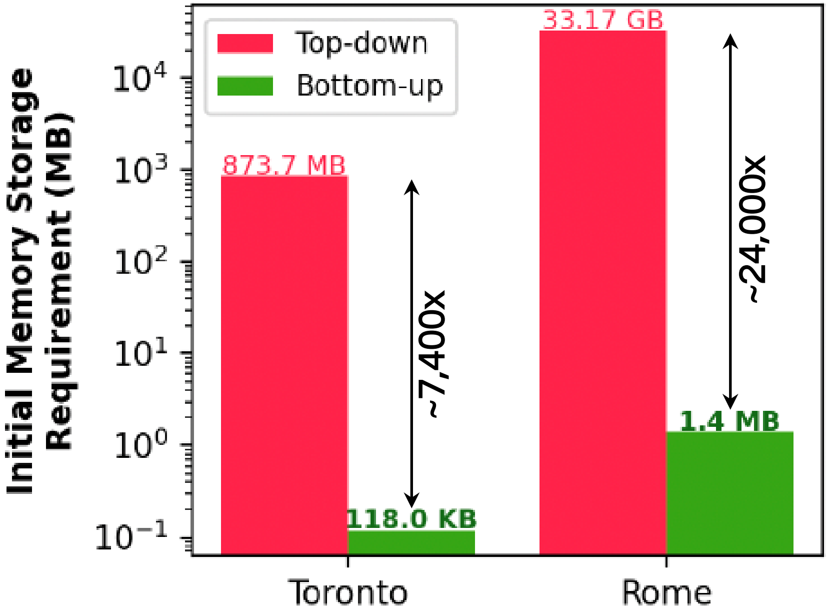
[1] Agarwal et al, PODS '18

[26] Chen et al, SIGSPATIAL '13

YORK U

RQ 2) Memory Efficiency

Bottom-up approaches outperform top-down methods

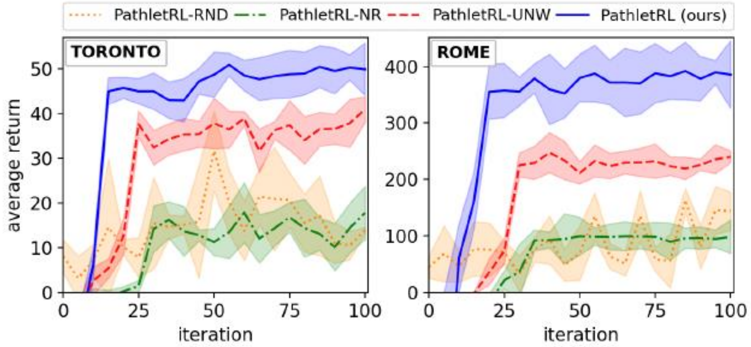


YORK U

RQ 3) Ablation Study – Average Returns

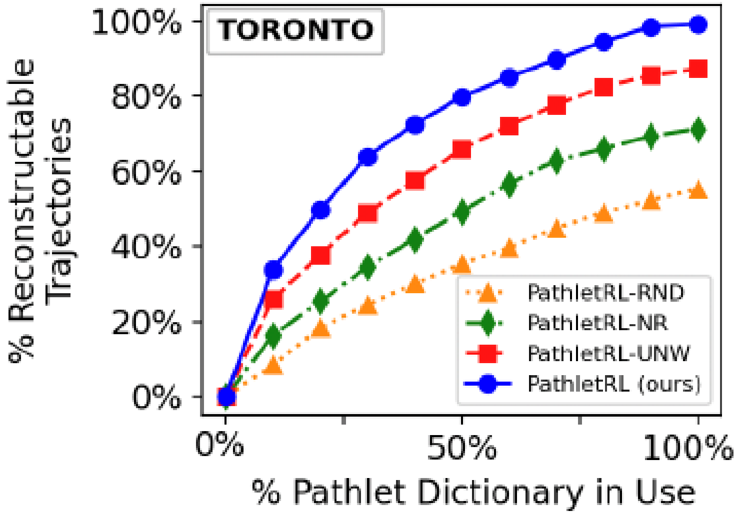
PathletRL Algorithm	Representability Measure	Weighted Networks	Deep Learning Policy
PathletRL-Nr	✗	✓	✓
PathletRL-Rnd	✓	✓	✗
PathletRL-Unw	✓	✗	✓
PathletRL (ours)	✓	✓	✓

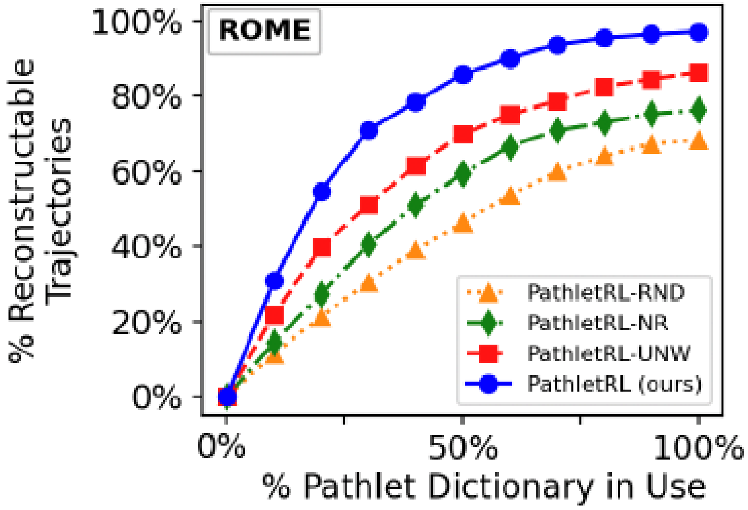
- PathletRL-RND has the poorest performance
    - Exhibits random RL policy (no learning)
    - All other methods converge after some iteration
- PathletRL-NR does not do well
    - Missing representability metric
- PathletRL-UNW is only a runner-up
    - Neglect the essence of pathlet weights
- PathletRL (ours) demonstrates the best performance



YORK U

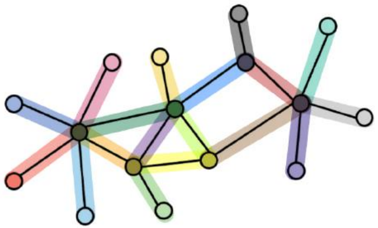
RQ 4) Partial Trajectory Reconstruction



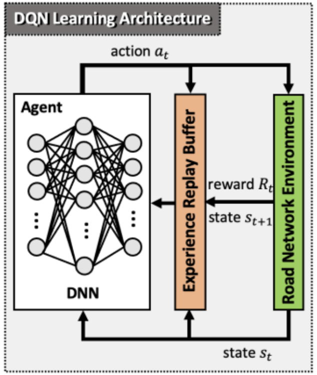


Conclusions

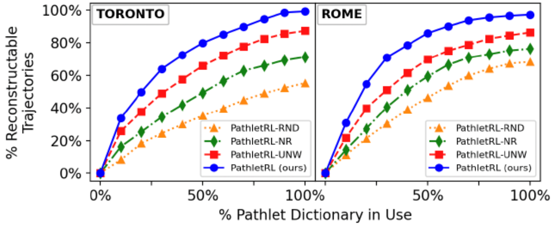
Take-away Message



Edge-disjoint pathlets



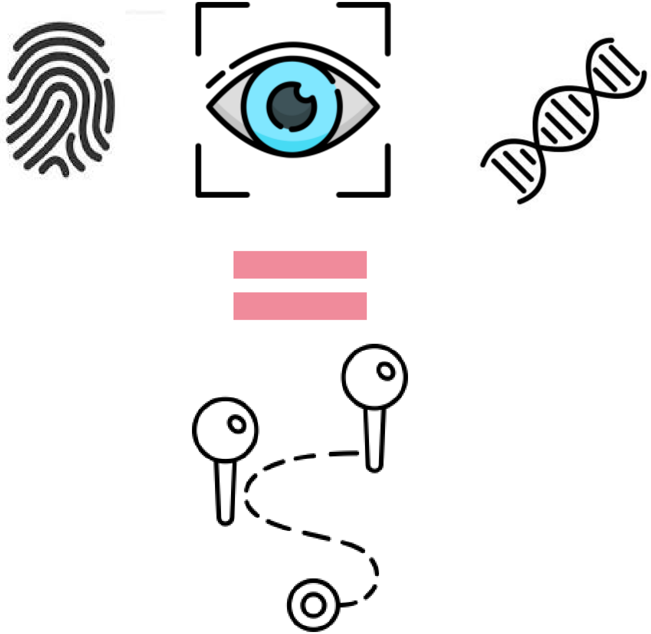
Deep Reinforcement Learning (DQN)



Partial trajectory reconstruction ~85%

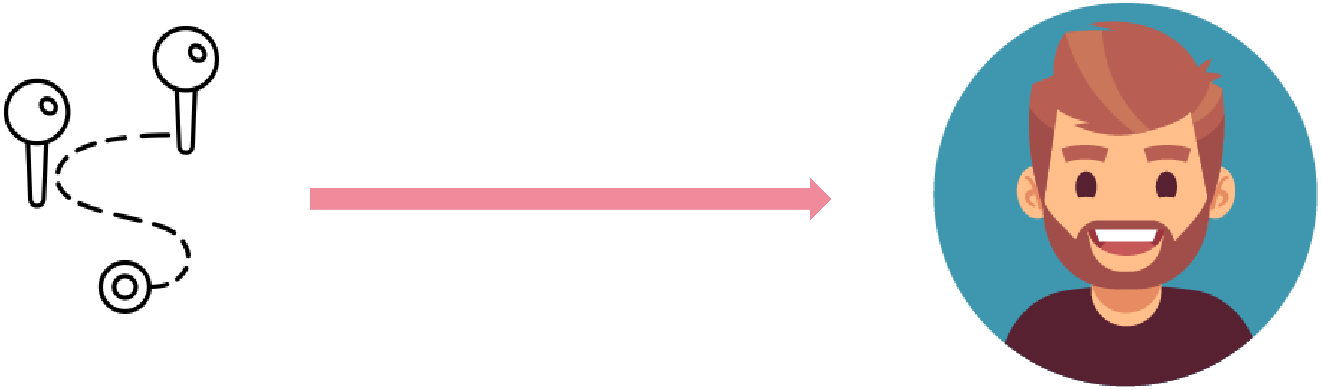
YORK U

Trajectory-User Linking using Higher-order Mobility Flow Representations



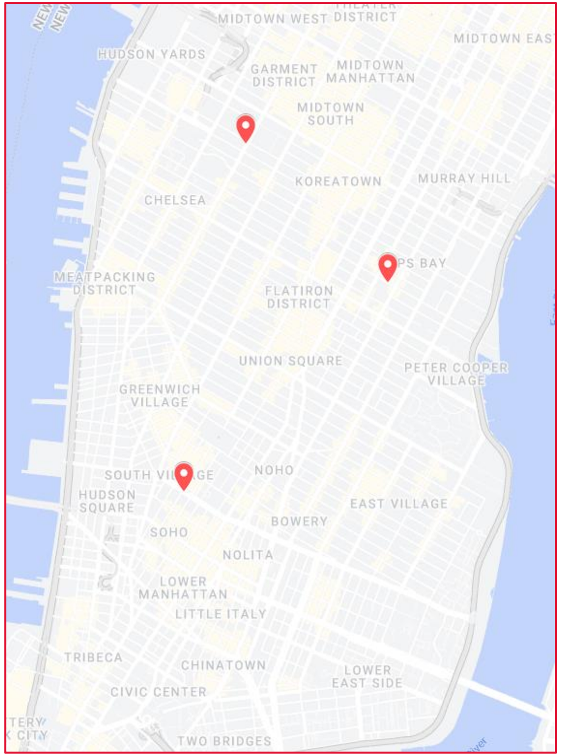
can trajectories
help to identify a person?

Trajectory-user Linking (TUL)



trajectory-user linking **aims at linking** anonymous trajectories to users who generate them

Data for Trajectory-user Linking (TUL)



Check-ins Trajectory

Mobility flow

YORK U

Limitations of the current approaches

Data Quality

- low accuracy and completeness

Data sparsity

- limited data

Imbalanced Data

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

Problem Definition

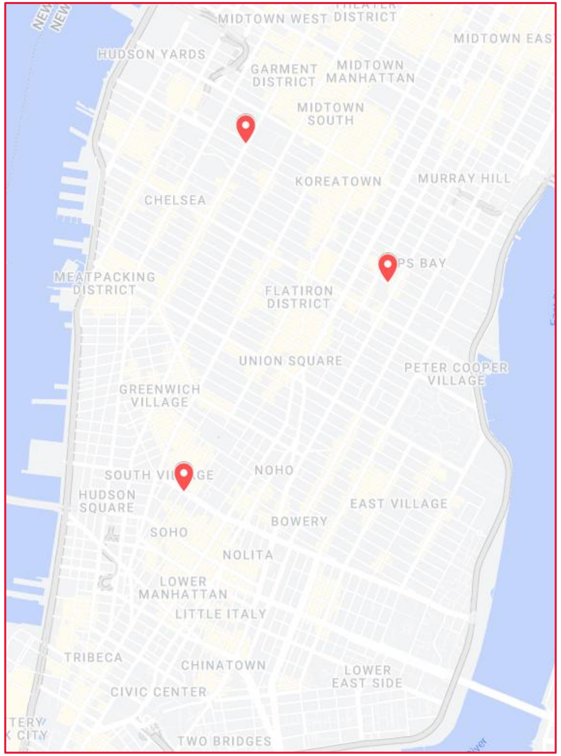
What is a check-in trajectory ?

Check-in record/visit

- $r = (u, p, t, \langle x, y \rangle)$

Check-in trajectory set

- $Tr = \{r_1, r_2, \ldots., r_m\}$



Check-ins Trajectory

YORK U

Problem Definition

Trajectory-user linking aims at linking anonymous trajectories to users

Given:

$\mathcal{U} = \{u_1, u_2, u_3, .., u_c\}$ – users

$\mathcal{T} = \{Tr_1, Tr_2, \dots, Tr_n\}$ – unlinked trajectories

TUL is defined as **a multiclass classification problem**

$$\min_{f \in \mathcal{F}} \mathbb{E}[\mathcal{L}(f(Tr_i), ui)] \; over \; \mathcal{F}$$

$$where\ \mathcal{F}\ \ is\ the\ set\ of\ all\ classifiers\ in\ the\ hypothesis\ space$$

$$\mathcal{L}(\cdot)\ is\ the\ loss\ between\ the\ predicted\ label\ f(Tr_i) \in \mathcal{U}\ and\ the\ true\ label\ u_i \in \mathcal{U}$$

YORK U

Methodology

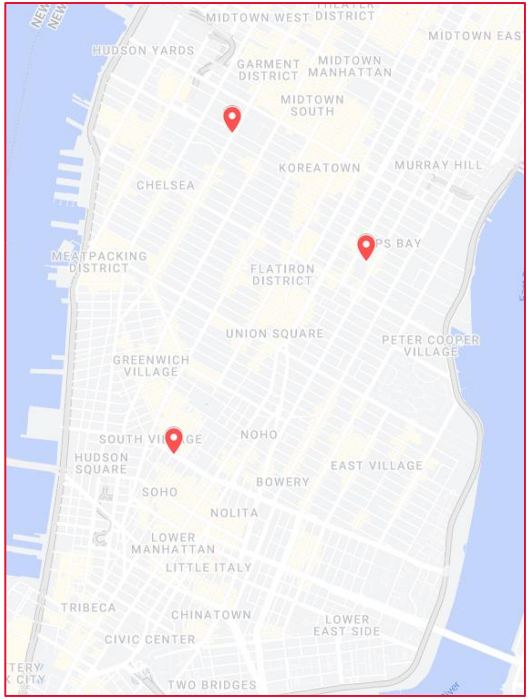
Overview

Step 1: Generating **higher-order mobility flow** representations

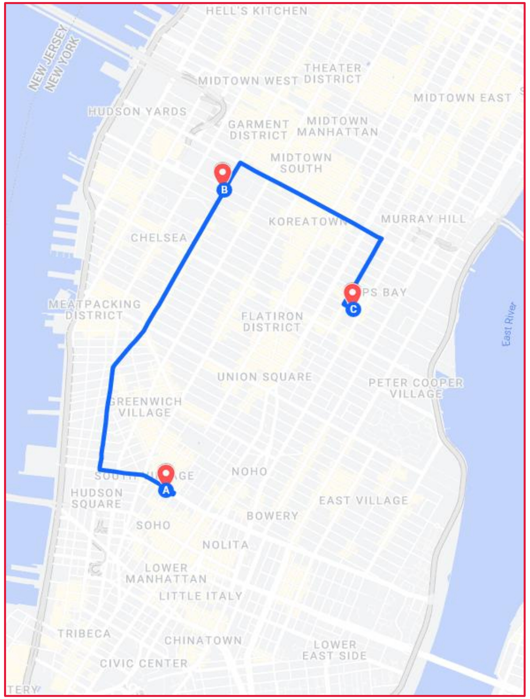
- generating **mobility flow** data from check-ins
- generating **higher-order mobility flow** and check-ins

Step 2: Modeling **trajectory-user linking**

Generating Mobility flow data



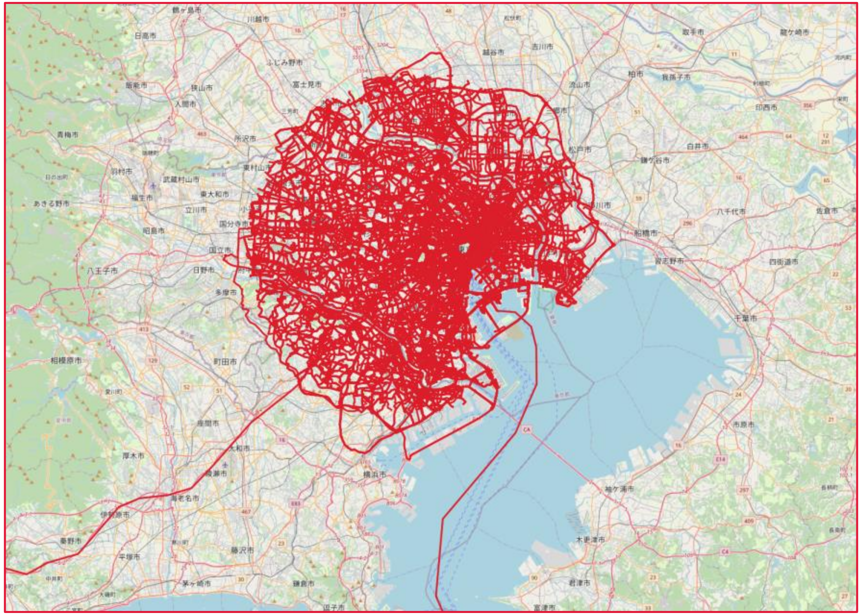
Check-ins Trajectory



Mobility flow

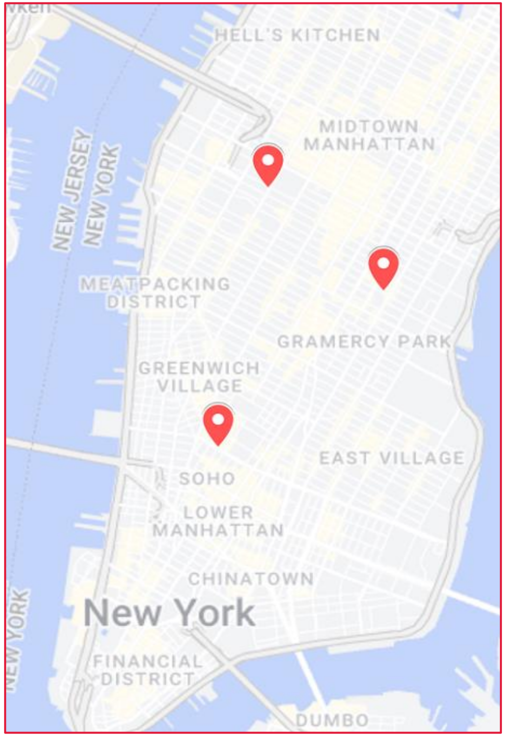
Mobility flow of NYC and TKY

NYC

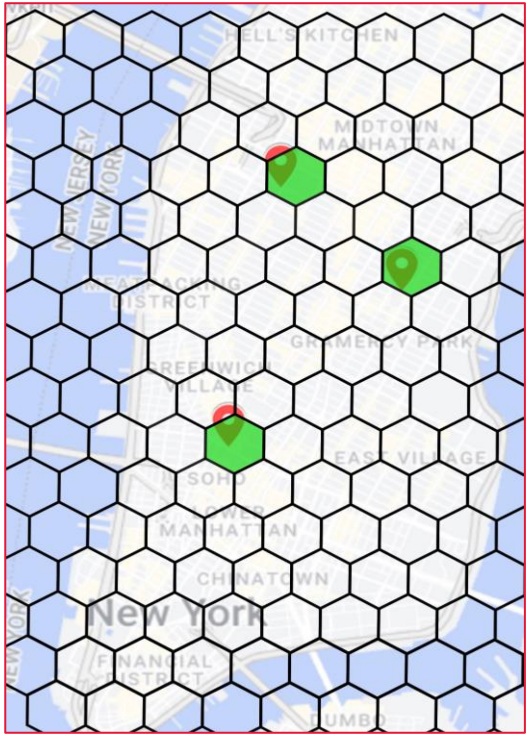


TKY

Generating higher-order check-ins



Check-ins Trajectory



Higher-order check-ins

YORK U

Translate check-ins to Higher-order

Check-ins

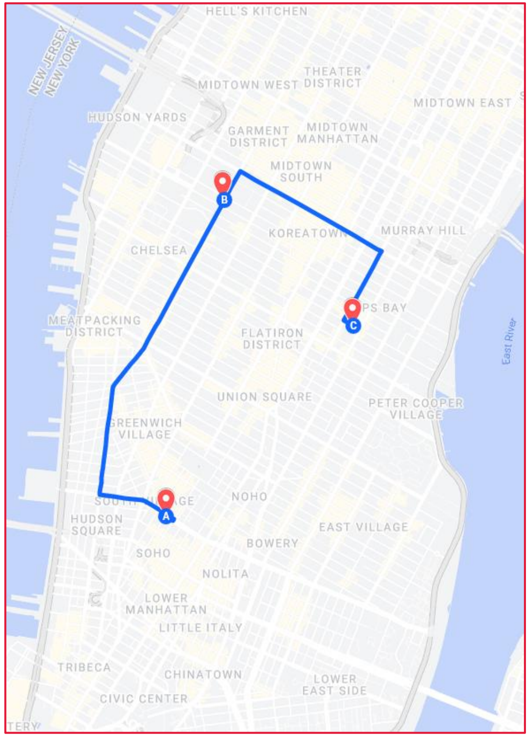
$$Tr = \{r_1, r_2, \dots., r_m\} = \{(p_1, t_1, \langle x_1, y_1 \rangle), (p_2, t_2, \langle x_2, y_2 \rangle), \dots, (p_m, t_m, \langle x_m, y_m \rangle)\}$$

Higher-order

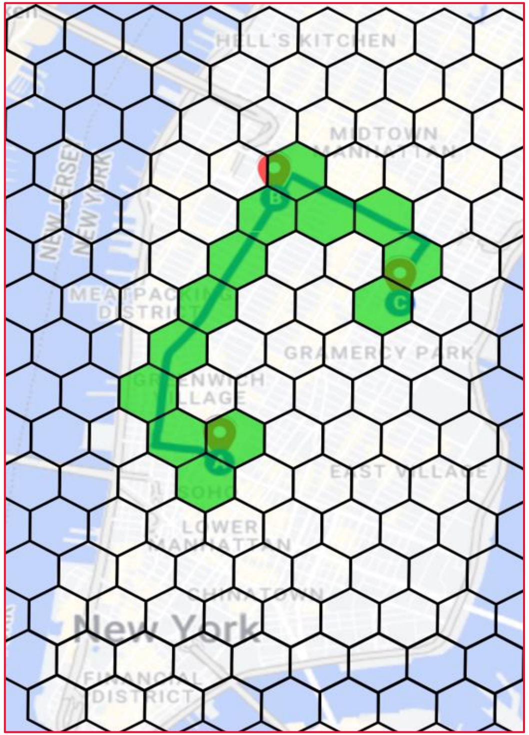
$$\{(p_1, t_1, \mathrm{g}_1), \quad (p_2, t_2, \mathrm{g}_2), \dots \dots., (p_m, t_m, \mathrm{g}_\mathrm{m})\}$$

Each trajectory now represents a sequence of continuous grid cells $\{g_1, g_2, \dots.\}$

Generating Higher-order Mobility flow



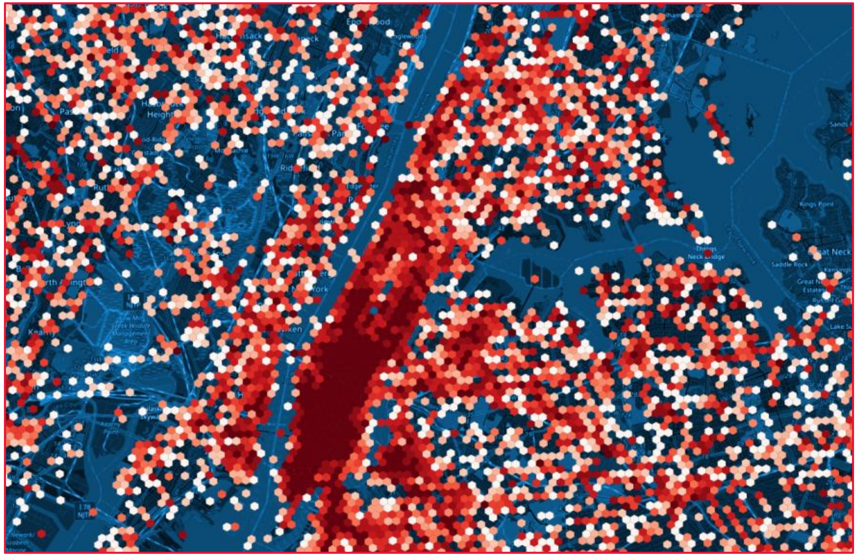
Mobility flow



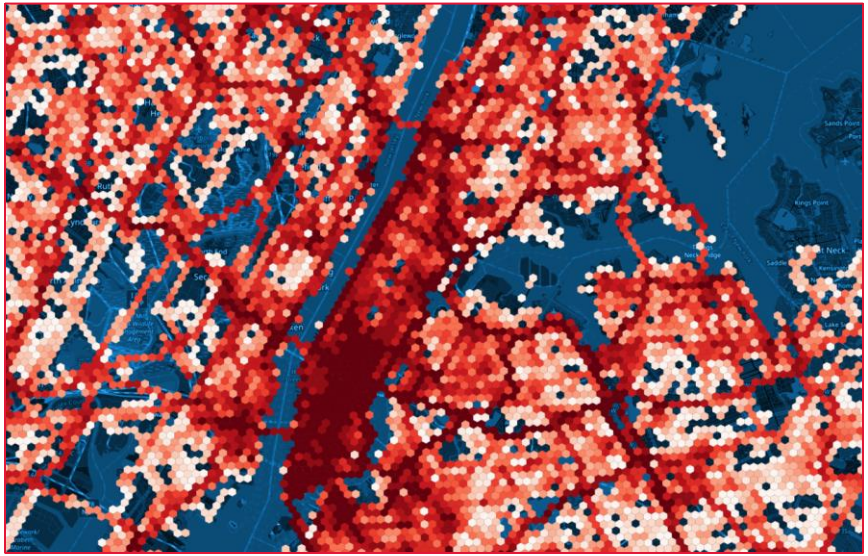
Higher-order
Mobility flow

YORK U

FOURSQUARE-NYC Heatmap



Higher-order check-ins



Higher-order Mobility flow

How to calculate Sparsity ?

Alex
$\{1,3\}$

Eve
$\{2\}$

Bob
$\{1,2,3\}$

$$\begin{array}{cc} & \begin{array}{ccc} p_1 & p_2 & p_3 \end{array} \\ \begin{array}{c} \text{Alex} \\ \text{Eve} \\ \text{Bob} \end{array} & \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \end{array}$$

Sparsity = % of zeros in User-POI matrix

$$\frac{3}{9} = 30\%$$

Higher-order Sparsity

Alex

{1,3}

$\{g1, g2\}$

Eve

{2}

$\{g2\}$

Bob

{1,2,3}

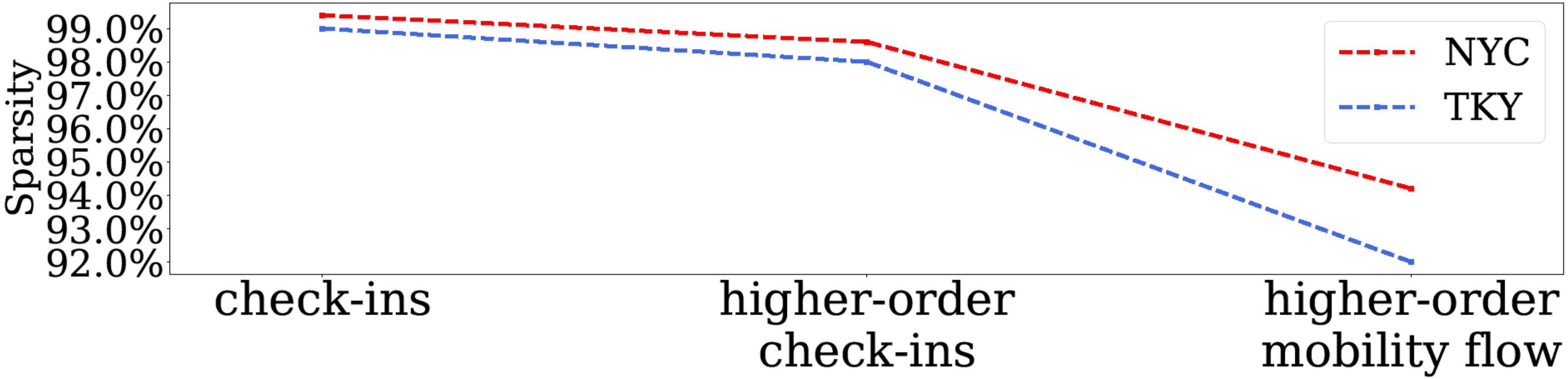
$\{g1, g2, g2\}$

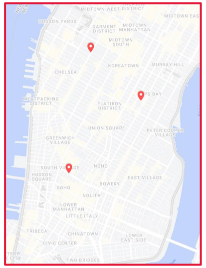
$$
\begin{array}{c}
 \\
\text{Alex} \\
\text{Eve} \\
\text{Bob}
\end{array}
\begin{array}{c}
\begin{array}{cc} g_1 & g_2 \end{array} \\
\begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 2 \end{bmatrix}
\end{array}
$$

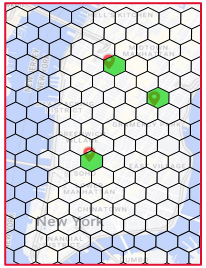
$$\frac{1}{6} = 16\%$$

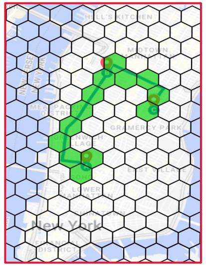
Check-in Sparsity $\geq$ Higher-order Sparsity

Impact of higher-order abstraction on sparsity









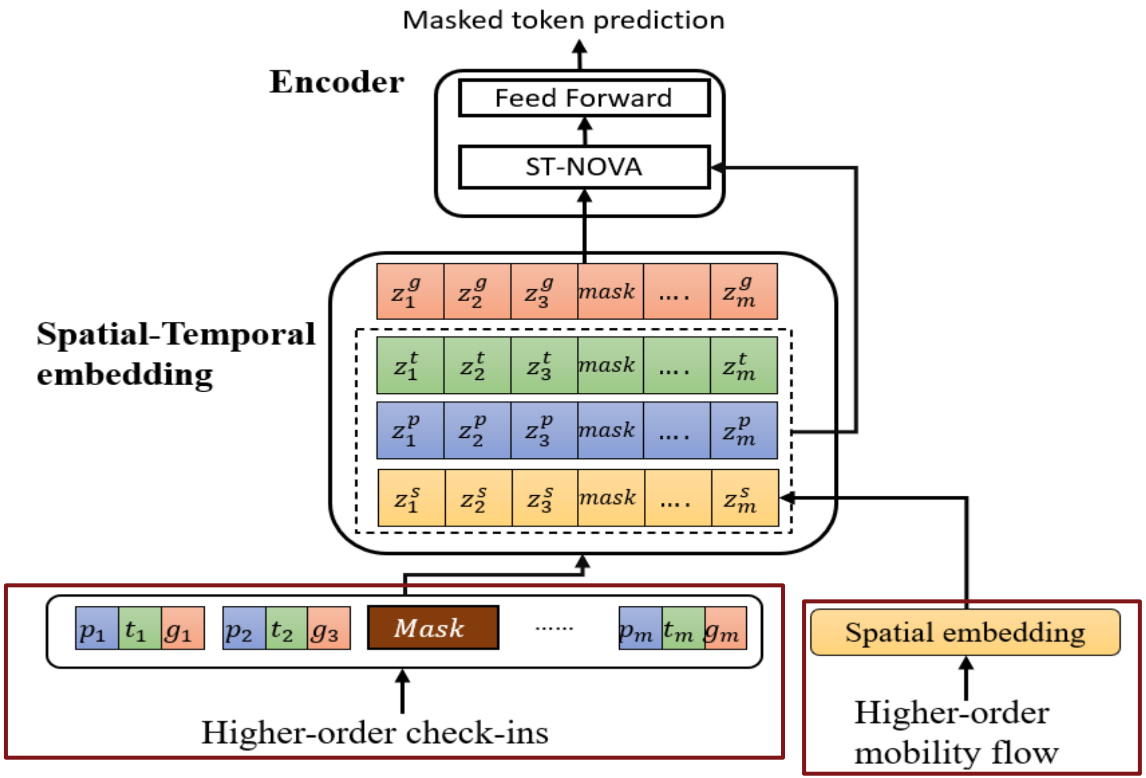
Overview

Step 1: Generating **higher-order mobility flow** representations

- generating **mobility flow** data from check-ins
- generating **higher-order mobility flow** and check-ins

Step 2: Modeling **trajectory-user linking**

TULHOR (trajectory-user linking using higher-order representations)



Two stages

Pre-training TULHOR

- **Input:** higher-order check-ins + masking, higher-order mobility flow
- **Output:** predicting masked token

Fine-tuning TULHOR

- **Input** : higher-order check-ins
- **Output:** user who generated the higher-order check-ins

Experiments

Overview

Datasets

- Foursquare NYC and TKY

Experiments

- TULHOR accuracy performance (vs SOTA and baselines)
- TULHOR Ablation study
- Tessellation granularity (grid size) effect

DATASET	$\|\mathcal{U}\|$	$\|\mathcal{T}\|$
	108	6795
FOURSQUARE-NYC	209	9,637
	234	10,133
	108	9343
FOURSQUARE-TKY	209	14,151
	451	20,964

Baselines

Conventional ML:

- Decision Tree
- Linear Discriminant Analysis (LDA)
- Linear Support Vector Machine (SVM)

TULER:

- RNN
- LSTM
- GRU

DeepTUL

- RNN (DeepTUL)
- LSTM (Attn-LSTM)
- GRU (Attn-GRU)

TULHOR performance (Foursquare TKY)

	Foursquare-TKY														
	$\vert\mathcal{U}\vert = 108$					$\vert\mathcal{U}\vert = 209$					$\vert\mathcal{U}\vert = 451$				
Model	Acc@1	Acc@5	P	R	F1	Acc@1	Acc@5	P	R	F1	Acc@1	Acc@5	P	R	F1
DT	0.789	0.793	0.785	0.777	0.775	0.658	0.664	0.629	0.615	0.613	0.522	0.525	0.446	0.437	0.431
LDA	0.853	0.912	0.927	0.847	0.874	0.722	0.808	0.778	0.692	0.713	0.574	0.720	0.553	0.501	0.495
LINEAR-SVM	0.890	0.948	0.923	0.886	0.898	0.769	0.878	0.794	0.736	0.748	0.609	0.761	0.610	0.539	0.550
TULER	0.870	0.933	0.871	0.860	0.860	0.768	0.864	0.762	0.735	0.736	0.637	0.74	0.588	0.554	0.548
TULER-L	0.905	0.952	0.904	0.898	0.897	0.848	0.911	0.837	0.825	0.824	0.739	0.827	0.708	0.675	0.675
TULER-G	0.915	0.954	0.916	0.910	0.909	0.851	0.911	0.842	0.824	0.825	0.738	0.823	0.701	0.672	0.671
Att-LSTM	0.908	0.966	0.916	0.901	0.908	0.752	0.871	0.795	0.729	0.760	0.407	0.584	0.362	0.326	0.343
Att-GRU	0.933	**0.975**	0.932	0.928	0.930	0.869	0.937	0.872	0.856	0.864	0.742	0.821	0.715	0.689	0.695
DeepTUL	0.922	0.966	0.927	0.913	0.920	0.773	0.904	0.820	0.747	0.782	0.660	0.790	0.631	0.587	0.608
TULHOR	**0.939**	0.973	**0.937**	**0.934**	**0.933**	**0.893**	**0.953**	**0.883**	**0.877**	**0.875**	**0.801**	**0.888**	**0.783**	**0.755**	**0.752**
Improvement	0.58%	-0.26%	0.59%	0.71%	0.37%	2.7%	1.77%	1.33%	2.53%	1.30%	7.86%	7.47%	9.52%	9.53%	8.11%

TULHOR outperforms every baseline

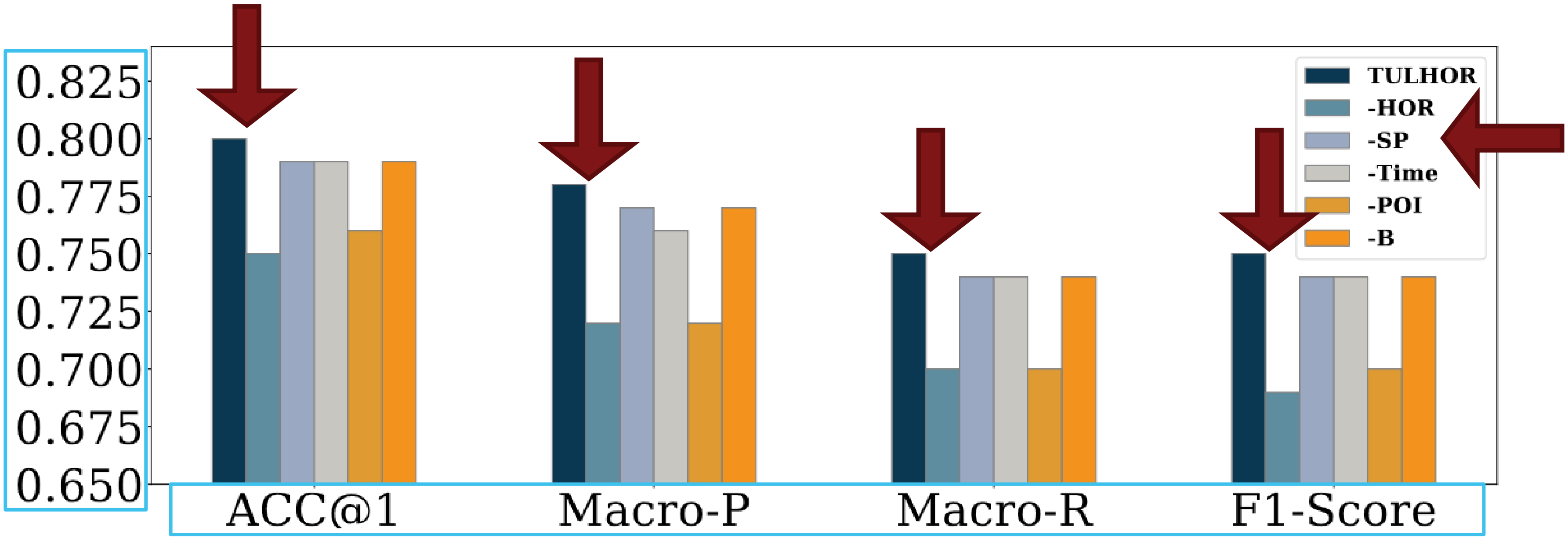
TULHOR has better scalability

YORK U

TULHOR performance (Foursquare NYC)

	Foursquare-NYC														
	$\lvert\mathcal{U}\rvert = 108$					$\lvert\mathcal{U}\rvert = 209$					$\lvert\mathcal{U}\rvert = 234$				
Model	Acc@1	Acc@5	P	R	F1	Acc@1	Acc@5	P	R	F1	Acc@1	Acc@5	P	R	F1
DT	0.884	0.892	0.878	0.867	0.868	0.785	0.788	0.753	0.728	0.730	0.778	0.782	0.722	0.712	0.705
LDA	0.822	0.851	0.962	0.810	0.868	0.746	0.781	0.791	0.687	0.718	0.696	0.752	0.724	0.615	0.650
LINEAR-SVM	0.873	0.929	**0.966**	0.878	0.909	0.776	0.839	0.785	0.702	0.727	0.731	0.798	0.724	0.628	0.657
TULER	0.870	0.929	0.869	0.851	0.852	0.776	0.853	0.749	0.722	0.718	0.768	0.844	0.733	0.707	0.703
TULER-L	0.903	0.942	0.904	0.890	0.890	0.847	0.898	0.828	0.803	0.807	0.845	0.889	0.821	0.806	0.803
TULER-G	0.909	0.949	0.914	0.897	0.898	0.854	0.892	0.835	0.811	0.812	0.846	0.891	0.821	0.805	0.803
Att-LSTM	0.823	0.896	0.715	0.703	0.709	0.716	0.832	0.554	0.559	0.556	0.712	0.830	0.569	0.557	0.563
Att-GRU	0.886	0.933	0.779	0.779	0.791	0.835	0.891	0.663	0.680	0.671	0.889	**0.936**	0.741	0.738	0.740
DeepTul	0.853	0.923	0.765	0.738	0.751	0.733	0.840	0.614	0.597	0.606	0.789	0.891	0.607	0.617	0.612
TULHOR	**0.940**	**0.966**	0.938	**0.931**	**0.932**	**0.903**	**0.943**	**0.890**	**0.877**	**0.876**	**0.892**	0.932	**0.876**	**0.864**	**0.860**
Improvement	3.42%	1.85%	-2.89%	3.85%	2.53%	5.82%	5.07%	6.58%	7.83%	7.87%	0.35%	-0.49%	6.61%	7.13%	7.19%

Ablation study

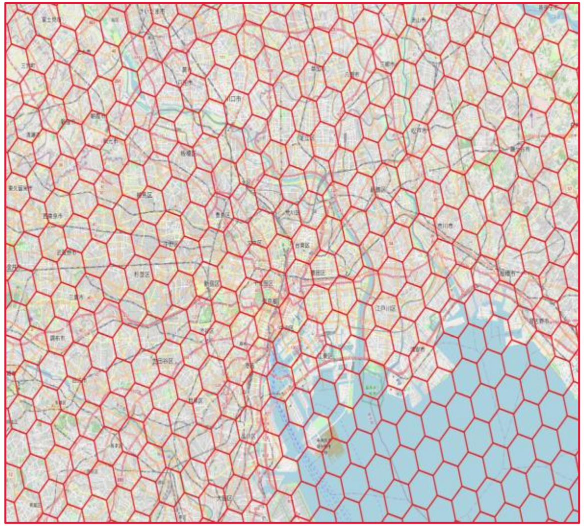


Removing Higher-order significantly reduces the performance

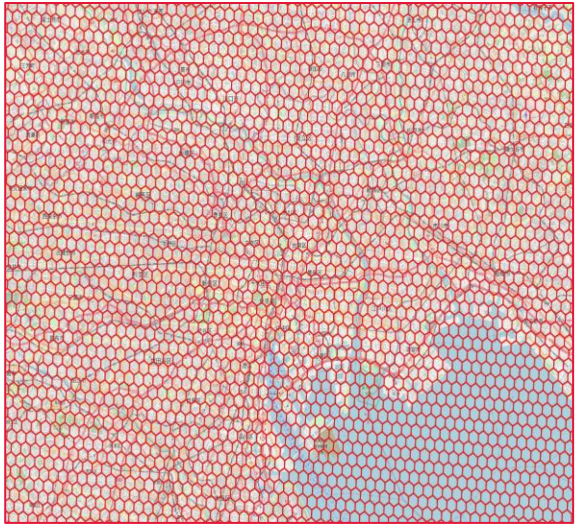
Tessellation granularity (grid size) effect

RESOLUTION	# OF CELLS	CELL SIZE (km^2)
HEX@7	334	5.160
HEX@8	2,003	0.730
HEX@9	11,036	0.015

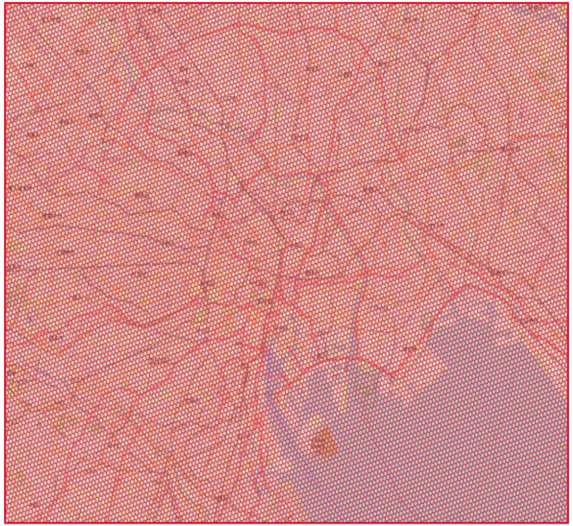
Tessellations of Tokyo



Hex@7



Hex@8



Hex@9

YORK U

Results of grid size study

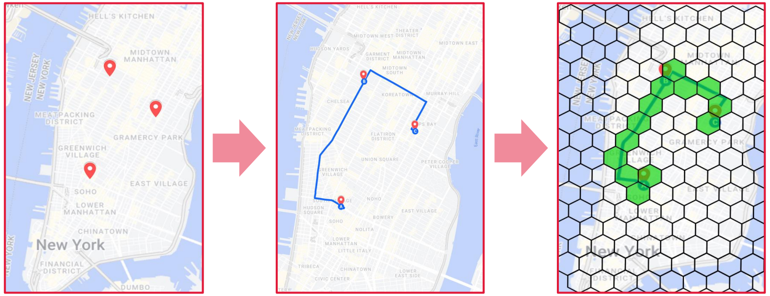
	FOURSQUARE-TKY														
	#USERS = 108					#USERS = 209					#USERS = 451				
METHOD	ACC@1	ACC@5	P	R	F1	ACC@1	ACC@5	P	R	F1	ACC@1	ACC@5	P	R	F1
HEX@7	0.923	0.971	0.920	0.911	0.913	0.868	0.943	0.832	0.817	0.815	0.711	0.883	0.734	0.734	0.711
HEX@8	0.926	**0.977**	0.925	0.917	0.917	0.868	0.940	0.862	0.849	0.849	0.790	0.884	0.753	0.740	0.733
HEX@9	**0.939**	0.973	**0.937**	**0.934**	**0.933**	**0.893**	**0.953**	**0.883**	**0.877**	**0.875**	**0.801**	**0.888**	**0.783**	**0.755**	**0.752**

Hex@9 outperforms other sizes as the number of users increases

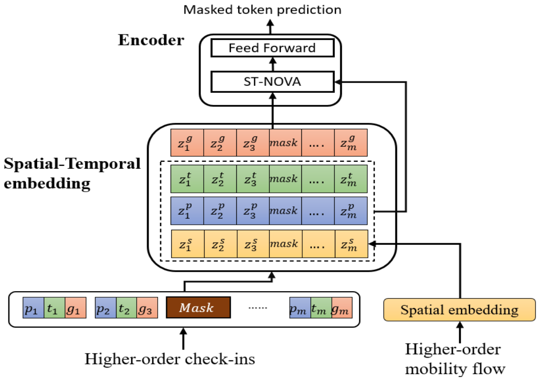
The smaller the cells are the better the scalability

Conclusions

Take-away Message



Higher-order mobility flow data generation



TULHOR: model for dealing with sparsity and low data quality of the TUL problem

YORK U

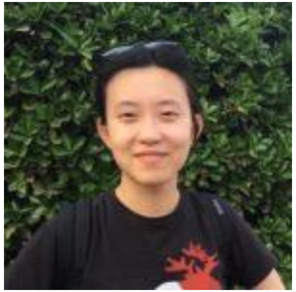
Questions

Credits

Gian Alix

Mahmoud Alsaeed

Ali Faraji



Jing Li

Nina Yanin

Amirhossein Nadiri

PathletRL: Trajectory Pathlet Dictionary Construction using Reinforcement Learning. G. Alix, M. Papagelis. **ACM SIGSPATIAL 2023** (In Press).

Trajectory-User Linking using Higher-order Mobility Flow Representations. M. Alsaeed, A. Agrawal, M. Papagelis. **IEEE MDM 2023**, pp. 158-167

Point2Hex: Higher-order Mobility Flow Data and Resources. A. Faraji, J. Ling, G. Alix, M. Alsaeed, N. Yanin, A. Nadiri, M. Papagelis. **ACM SIGSPATIAL 2023** (In Press).

Thank you!