Leveraging Deep Learning Methods for Trajectory Similarity Learning and Trajectory Pathlet Dictionary Construction

MSc. Thesis of **Gian Carlo Alix**

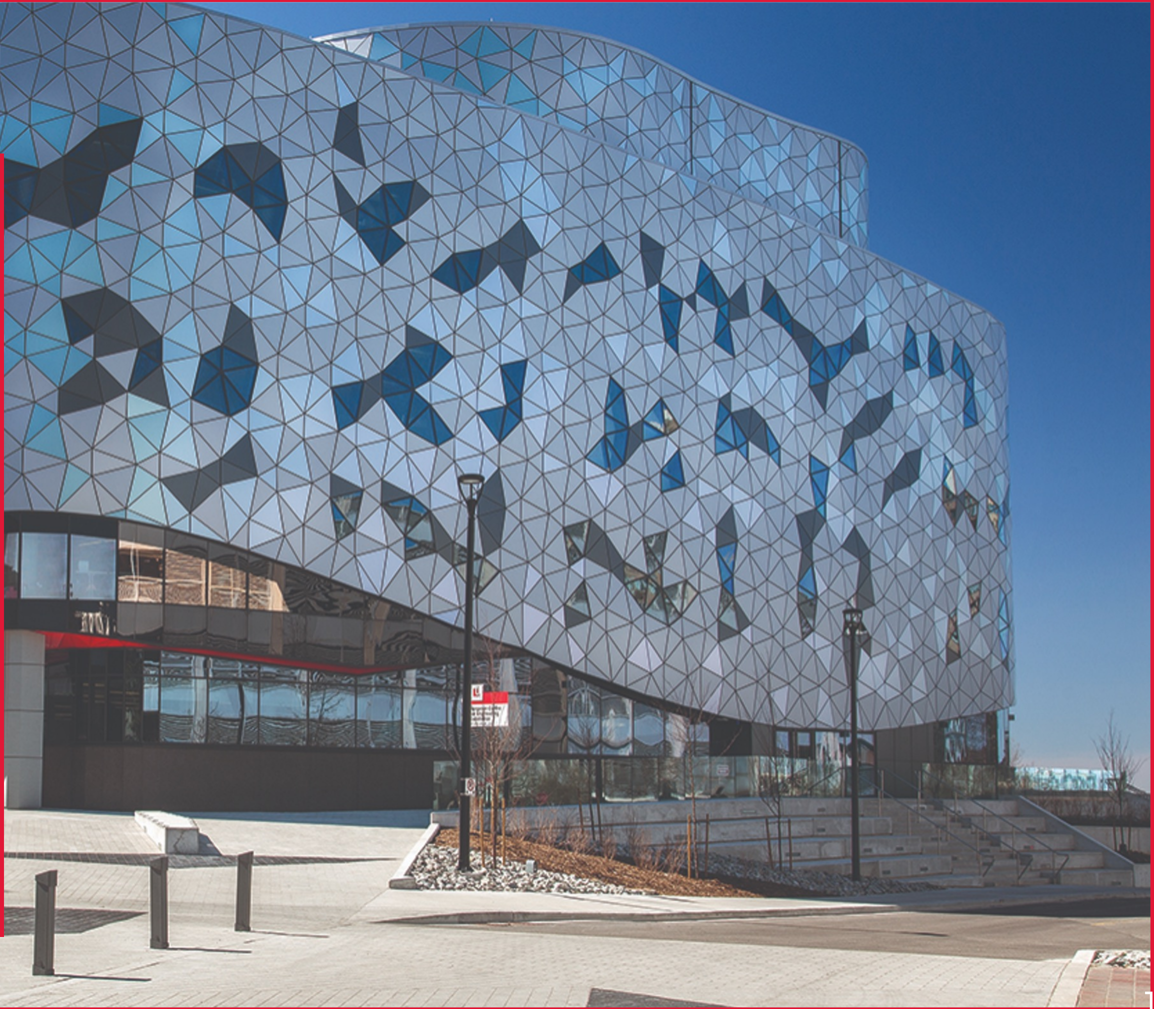
Department of Electrical Engineering and Computer Science

Committee Members:

Dr. Manos Papagelis (Supervisor)
Dr. Hina Tabassum (Chair)
Dr. Mehdi Nourinejad (Internal member)

Master's Oral Examination
August 3rd, 2023

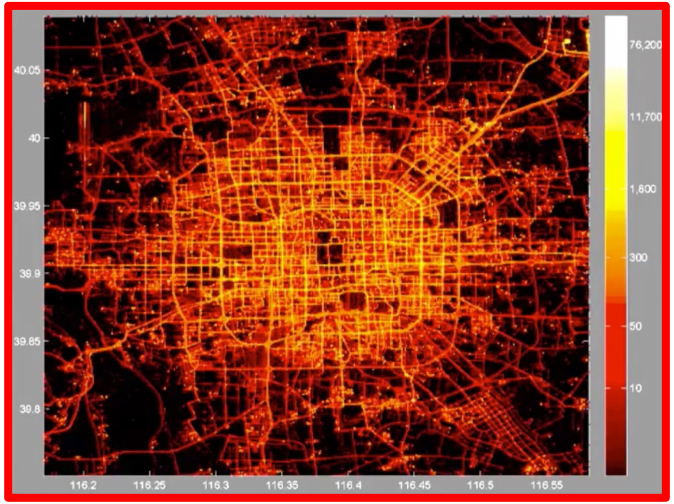
YORK U



Introduction

Motivation

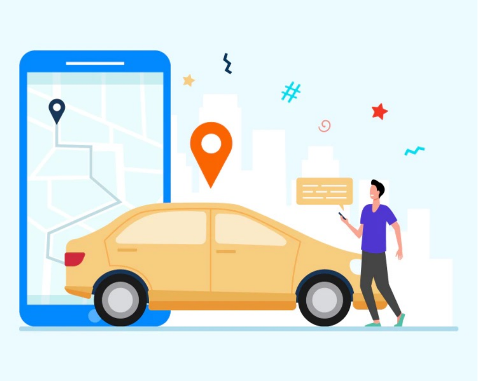
- Proliferation of geo-positioning capabilities
- Massive spatiotemporal trajectories of moving objects are collected
- Motivates various trajectory analytics



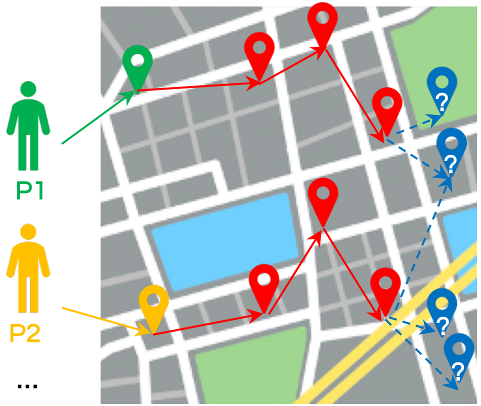
Trajectories contained within the 5th Ring Road in Beijing

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

Motivation

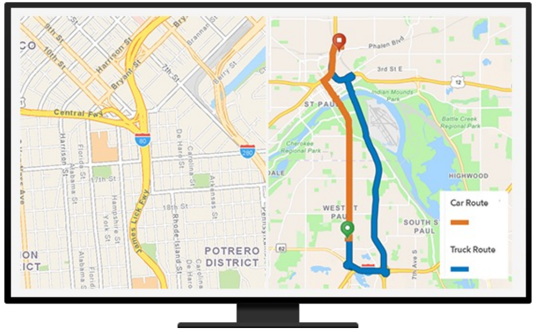


ridesharing



trip/POI (point-of-interest) recommendation

traffic analysis



route planning and optimization

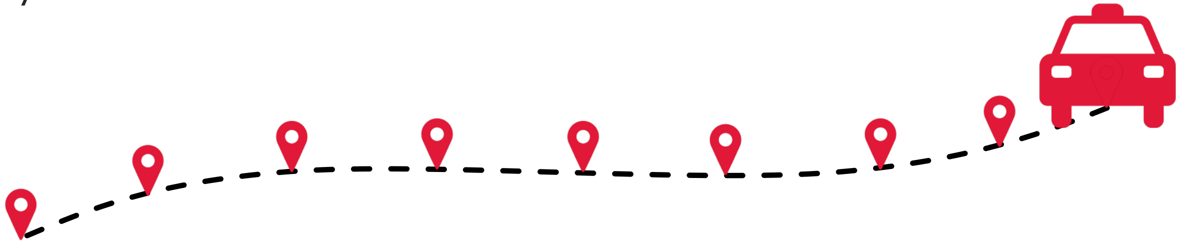
Trajectories on the Road Network

Trajectories

- Trajectory
  - Denoted by τ
  - Represented as:

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

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



Road Networks

- 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)

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

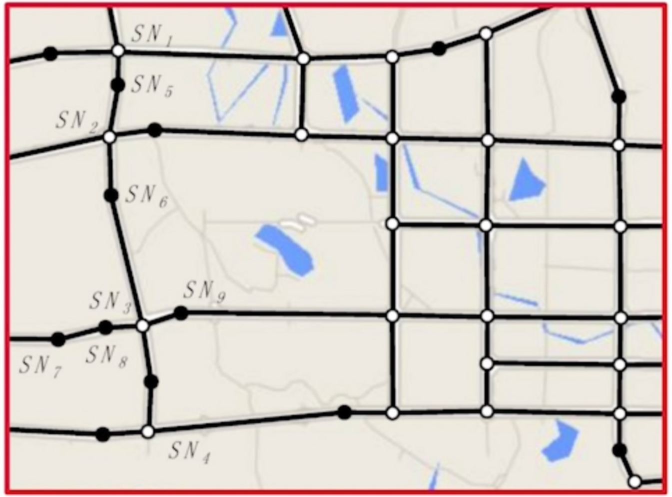
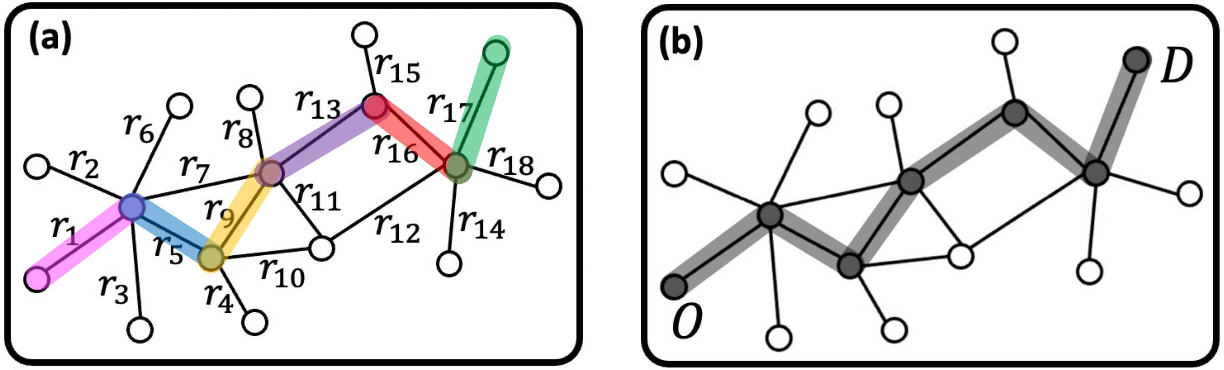


Image Source: “Updating Road Networks by Local Renewal from GPS Trajectories” [Wu et al, MDPI ‘16]

[†] “Road”, “segment” and “road segment” are terms used interchangeably.

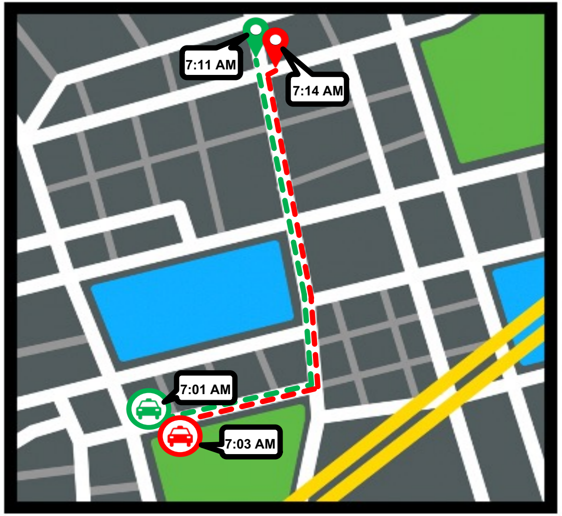
Road Segment-based Representation

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



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

Problems of Interest



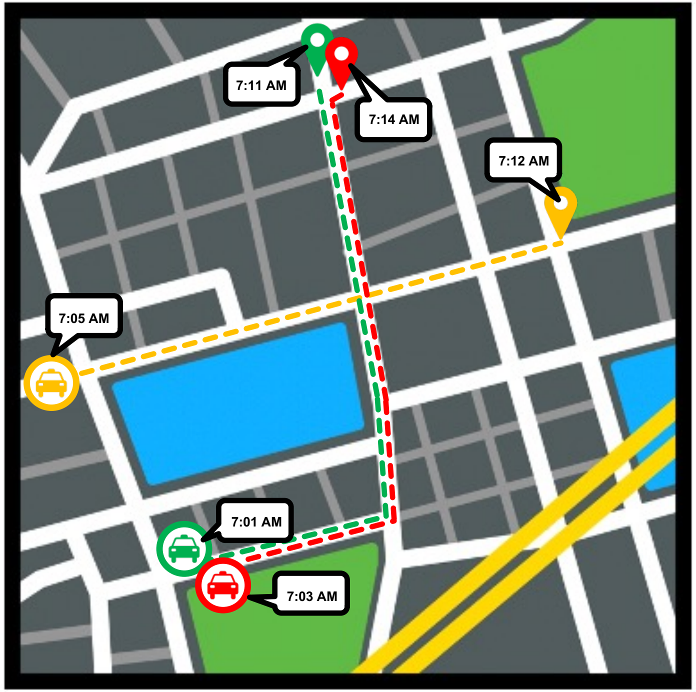
Trajectory Similarity Learning

Pathlet Dictionary Construction

Similarity in Trajectories

Trajectory Similarity

- How similar two trajectories are
- Several ways to define



Spatiotemporal Similarity

- The dot product of their spatiotemporal representations

$$\mathcal{S}(\tau_i, \tau_j) = \mathbf{x}_j^{\top}\mathbf{x}_i$$

- Spatiotemporal Weight

$$\boxed{\mathcal{S}(\tau_i, \tau_j) = \theta \cdot \mathcal{S}^{(s)}\left(\tau_i^{(s)}, \tau_j^{(s)}\right) + (1-\theta) \cdot \mathcal{S}^{(t)}\left(\tau_i^{(t)}, \tau_j^{(t)}\right)}$$

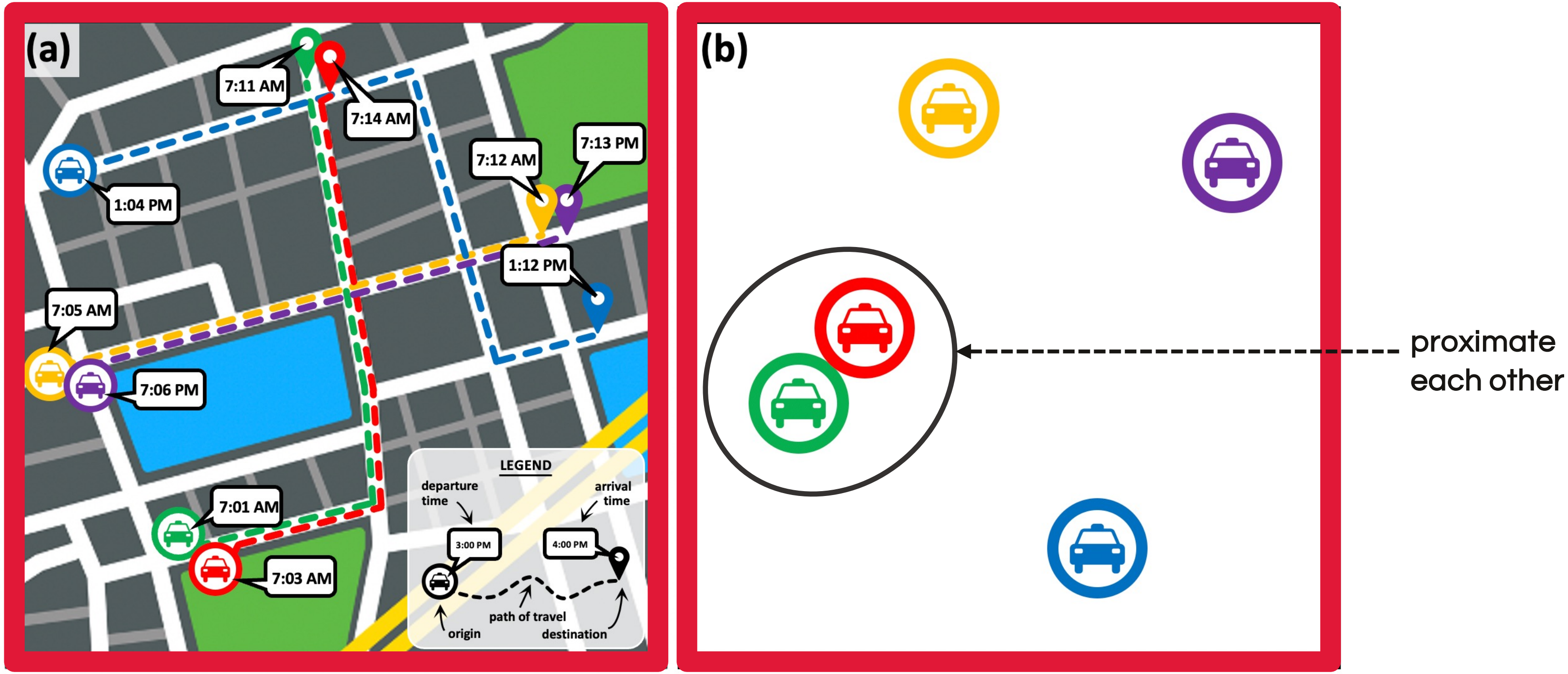
where

$\mathcal{S}(\cdot,\cdot)$ – network-aware distance measures

τ_i, τ_j – trajectories

$\theta \in [0,1]$ – control parameter

Spatiotemporal Similarity – Example



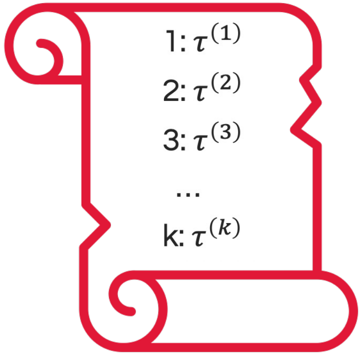
Taxi Trajectories

Embedding Space

Problem Statement

Problem Statement

- Top-k Trajectory Similarity Search Task
  - **Given:** Trajectory set $\mathcal{T}$
    Query trajectory τ_q
    Positive integer $k \geq 1$
  - **Find** the (ranked) list of top k trajectories in $\mathcal{T}$:
  - **Criterion:** Similarity with τ_q



Existing Works

Traditional Measures

- Free-Space Measures
  - Consider the geometric aspect of trajectories on the continuous, Euclidean space
  - Examples: DTW [Yi et al, ICDE '98], LCSS [Vlachos et al, ICDE '02], ERP [Chen et al, VLDB '04]
- Network-aware Measures
  - Consider properties of underlying network (e.g., network structure, connectivity, etc.)
  - Examples: TP [Shang et al, VLDB '17], DITA [Shang et al, SIGMOD '18], NetERP [Koide et al, VLDB '20]

pointwise matching ❯ quadratic-level complexity ❯ high computational cost ❯ bottleneck in massive trajectory datasets

Related Work and Limitations - 1

- Related Work
  - Use NN-based models for learning representations of trajectories
    [Li et al – ICDE '18, Yao et al – ICDE '19, Zhang et al – IJCAI '20, Han et al – KDD '21]
  - Similarity relations between trajectories are preserved in low-dimensional embeddings
  - Speedups compared to methods that operate directly on trajectories

- Limitations
  - Disregard the temporal dimension
  - Not effective in time-aware applications
    (e.g., transportation planning, monitoring, etc.)

Related Work and Limitations - 2

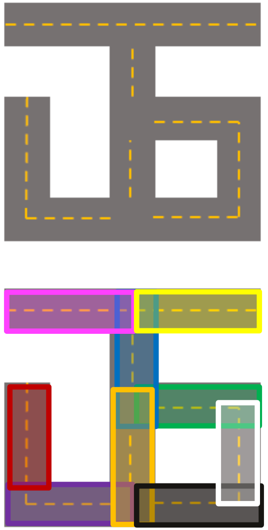
- Related Work
  - The temporal aspect of trajectories is important
    [Fu et al – TIST ‘20, Fang et al – KDD ‘22]
  - Timestamps considered in the decoding process
  - Captures the temporal regularities and periodic patterns in trajectories

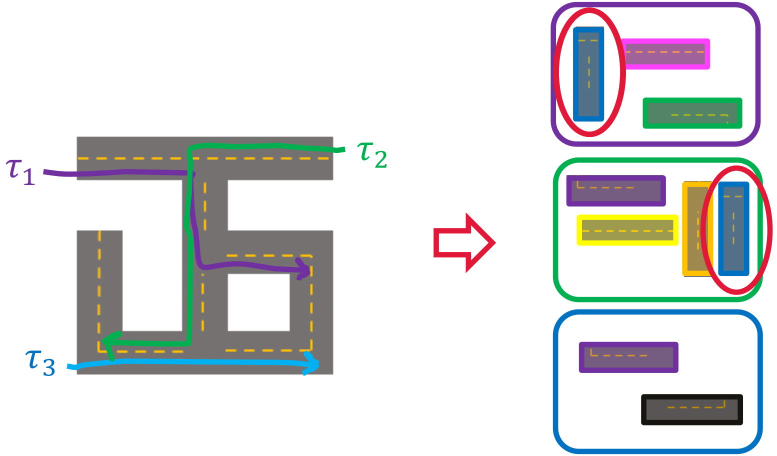
- Limitations
  - Performance of models can still be improved
  - Refined representations of trajectories

Methodology – ST2Box

Reducing Trajectory Similarity to Set Similarity Problem

- Treat each trajectory as a set; its elements are the road segments it has traversed (road-based representation $\mathfrak{R}(\tau)$)
- Similar (Dissimilar) trajectories map to similar (dissimilar) sets

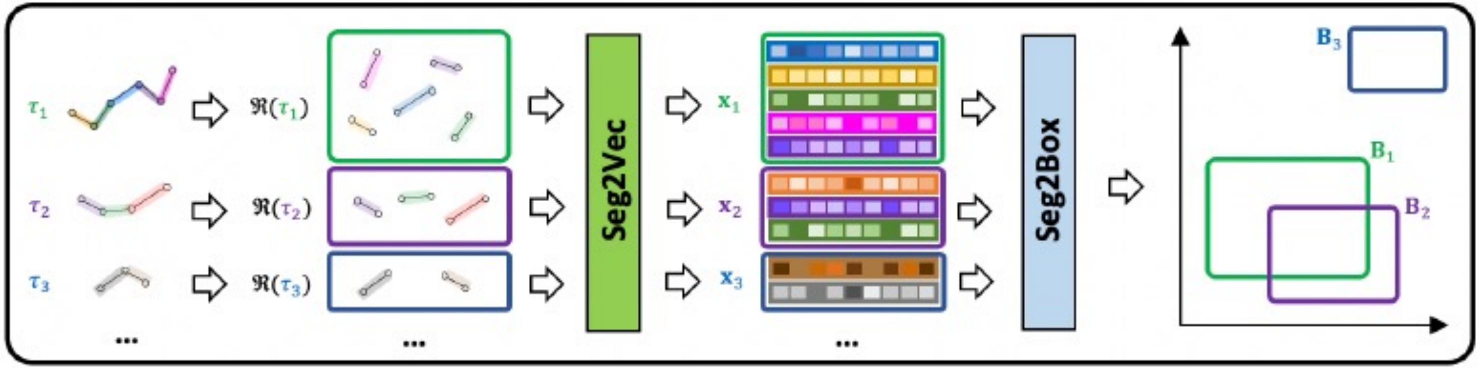




Trajectories τ_1 and τ_2 are similar!

ST2Box Overview

- Spatiotemporal Trajectories to Box Embeddings for Similarity Learning



- Seg2Vec – spatiotemporal vector representations of road segments
- Seg2Box – box representations of sets of road segments

YORK U

High Level Algorithm – ST2Box

Input: Trajectory set $\mathcal{T}$

Query trajectory τ_q

Positive integer $k \geq 1$

Output: the (ranked) list of top k trajectories in $\mathcal{T}$

1. Initialization
2. Spatiotemporal representation of each road segment
3. Expressing each trajectory as a set of road segments
4. Box representations of trajectories (sets)
5. Computing each trajectory's similarity score with the query
6. Ranking each trajectory by the similarity score criterion

$D_{vec} \leftarrow \textsc{Dict}()\,;\ S \leftarrow \textsc{Dict}()\,;\ S_{score} \leftarrow \textsc{Dict}()$

foreach $r \in \mathcal{E}$ **do**

  $\mathbf{z} \leftarrow \textsc{Seg2Vec}(r)$

  $D_{vec}[r] \leftarrow \mathbf{z}$

foreach $\tau \in \mathcal{T} \cup \{\tau_q\}$ **do**

  $R \leftarrow \varnothing$

  foreach $r \in \mathfrak{R}(\tau)$ **do**

    $R \leftarrow R \cup \{D_{vec}[r]\}$

  $S[\tau] \leftarrow R$

$\mathbf{B}_{model} \leftarrow \textsc{Seg2Box}(S \setminus \{\tau_q\})$

$\mathbf{q} \leftarrow \mathrm{ENCODE}(\mathbf{B}_{model}, S[\tau_q])$

foreach $\tau \in S \setminus \{\tau_q\}$ **do**

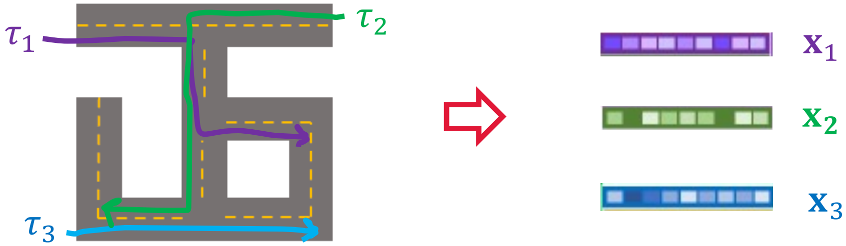
  $\mathbf{t} \leftarrow \mathrm{ENCODE}(\mathbf{B}_{model}, S[\tau])$

  $S_{score}[\tau] \leftarrow \mathcal{S}(\mathbf{t}, \mathbf{q})$

$\mathrm{SORT}(S_{score}.\mathrm{ITEMS}(), \mathrm{KEY} = \mathrm{LAMBDA}\ \mathbf{x} : \mathbf{x}[1])$

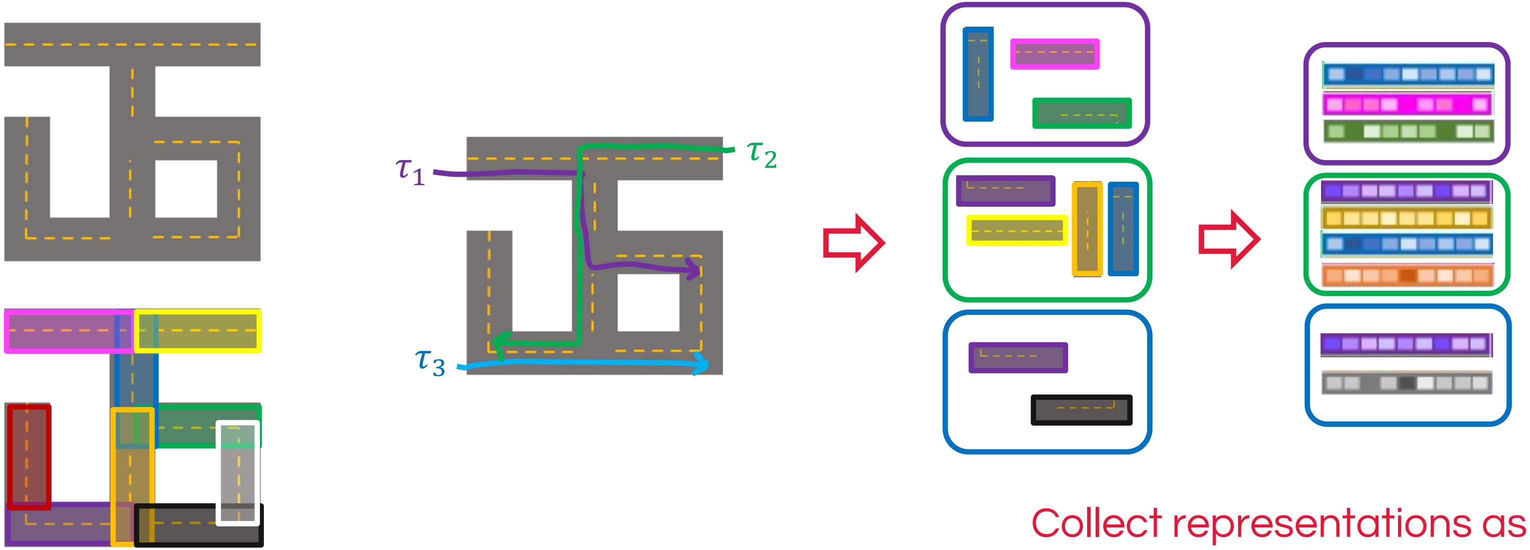
Paradigm: Trajectories to Vectors

- Method used by [Fang et al – KDD '22]



Seg2Vec Idea

- Adapt a similar method as [Fang et al – KDD '22] but with road segments instead of each individual trajectory
- More refined representations of trajectories



Collect representations as **sets**!

YORK U

Seg2Vec Components

- **Segment-based Spatial Representation Learning (SegSRL)**
  - Node2Vec (learning of adjacent roads)
  - Graph Convolution Networks (local smoothing)
  - LSTM with self-attentions (some roads are more important)

$$\tilde{\mathbf{r}} = \mathrm{GCN}(\mathbf{r}) = \sigma\left(\left(\sum_{r' \in \mathcal{N}(r)} a \cdot W_d \mathbf{r}'\right) \otimes \mathbf{r}\right)$$

- **Segment-based Temporal Representation Learning (SegTRL)**
  - Time-embedding

$$\hat{\mathbf{t}}[i] = \begin{cases} \xi_i t + \zeta_i, & i = 0 \\ \cos(\xi_i t + \zeta_i), & 1 \leq i \leq T \end{cases}$$

  - LSTMs (for learning temporal dependence)

- **Segment-based Spatiotemporal Fusion (SegSTF)**

$$\mathbf{Z}^{(s)} = W_F \mathbf{z}^{(s)} \text{ and } \mathbf{Z}^{(t)} = W_F \mathbf{z}^{(t)}$$

$$w(i,j) = \frac{\exp\left(W_Q \mathbf{Z}^{(i)} \cdot W_K \mathbf{Z}^{(j)}\right)}{\sum_{* \in \{s,t\}} \exp\left(W_Q \mathbf{Z}^{(i)} \cdot W_K \mathbf{Z}^{(*)}\right)}$$

$$\hat{\mathbf{z}}^{(s)} = \textsc{Norm}\left(\textsc{Ffn}\left(w(s,t)\mathbf{Z}^{(t)} \otimes w(s,s)\mathbf{Z}^{(s)}\right) \otimes \mathbf{z}^{(s)}\right)$$

$$\hat{\mathbf{z}}^{(t)} = \textsc{Norm}\left(\textsc{Ffn}\left(w(t,t)\mathbf{Z}^{(t)} \otimes w(t,s)\mathbf{Z}^{(s)}\right) \otimes \mathbf{z}^{(t)}\right)$$

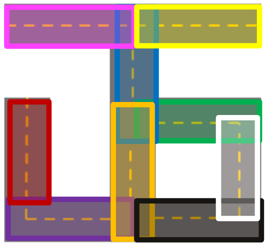
$$\Rightarrow \quad \mathbf{z} = \textsc{Lstm}\left(\hat{\mathbf{z}}^{(s)} \otimes \hat{\mathbf{z}}^{(t)}\right)$$

Neural Networks for Sets

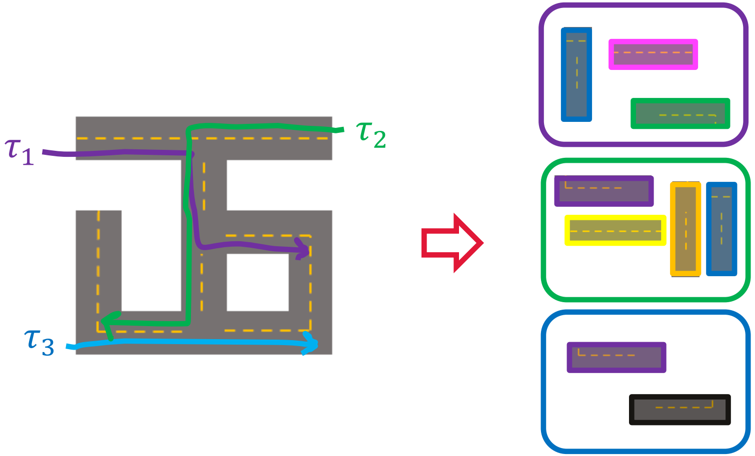
Neural Network Architecture for Sets

- Preserve similarity relations such as:
  - Overlapping coefficient
  - Cosine similarity
  - Jaccard index
  - Dice index
- Desire: similar trajectories to have similar set representations in the latent space

Seg2Box: A Set-to-Box Architecture for Set Representations



- Based on Set2Box [Lee et al – ICDM '22]
  - Applies to any type of set and elements
- We adapt Set2Box
  - Trajectories as sets
  - Road segments as elements
- Characteristics:
  - accurate
  - versatile
  - generalizable



The Box Architecture - Background

- Represent sets as hyperrectangles (boxes)

- Denote by box $\mathbf{B}_X$ of the road-based representation $\mathfrak{R}(\tau_X)$ for trajectory τ_X

- Targets:
  - Approximate $\mathbb{V}(\mathbf{B}_X) \propto |\mathfrak{R}(\tau_X)|$
  - Preserving similarity relations:
    - Approximate $\mathbb{V}(\mathbf{B}_X \cap \mathbf{B}_Y) \propto |\mathfrak{R}(\tau_X) \cap \mathfrak{R}(\tau_Y)|$

The Box Architecture – Objective

- Preseve elemental relationships between (triplets of) trajectories
- Consider the various cardinalities

Cardinalities	Measures
$C_1(\mathfrak{R}(\tau_X), \mathfrak{R}(\tau_Y), \mathfrak{R}(\tau_Z))$	$\lvert\mathfrak{R}(\tau_X)\rvert$
$C_2(\mathfrak{R}(\tau_X), \mathfrak{R}(\tau_Y), \mathfrak{R}(\tau_Z))$	$\lvert\mathfrak{R}(\tau_Y)\rvert$
$C_3(\mathfrak{R}(\tau_X), \mathfrak{R}(\tau_Y), \mathfrak{R}(\tau_Z))$	$\lvert\mathfrak{R}(\tau_Z)\rvert$
$C_4(\mathfrak{R}(\tau_X), \mathfrak{R}(\tau_Y), \mathfrak{R}(\tau_Z))$	$\lvert\mathfrak{R}(\tau_X) \cap \mathfrak{R}(\tau_Y)\rvert$
$C_5(\mathfrak{R}(\tau_X), \mathfrak{R}(\tau_Y), \mathfrak{R}(\tau_Z))$	$\lvert\mathfrak{R}(\tau_X) \cap \mathfrak{R}(\tau_Z)\rvert$
$C_6(\mathfrak{R}(\tau_X), \mathfrak{R}(\tau_Y), \mathfrak{R}(\tau_Z))$	$\lvert\mathfrak{R}(\tau_Y) \cap \mathfrak{R}(\tau_Z)\rvert$
$C_7(\mathfrak{R}(\tau_X), \mathfrak{R}(\tau_Y), \mathfrak{R}(\tau_Z))$	$\lvert\mathfrak{R}(\tau_X) \cap \mathfrak{R}(\tau_Y) \cap \mathfrak{R}(\tau_Z)\rvert$

- Sample positive ($\mathcal{T}^+$) and negative ($\mathcal{T}^-$) trajectory triplets
- The objective function is:

$$\boxed{\sum_{\{\tau_X,\tau_Y,\tau_Z\}\in\mathcal{T}^+\cup\mathcal{T}^-} \sum_{i=1}^{7} \left(\frac{C_i(\tau_X,\tau_Y,\tau_Z)}{\sum_{j=1}^{7} C_j(\tau_X,\tau_Y,\tau_Z)} - \frac{\hat{\mathbb{V}}_i(\mathrm{B}_X,\mathrm{B}_Y,\mathrm{B}_Z)}{\sum_{k=1}^{7} \hat{\mathbb{V}}_k(\mathrm{B}_X,\mathrm{B}_Y,\mathrm{B}_Z)} \right)^2}$$

where $\hat{V}_i$ is the approximated volume of the boxes corresponding to C_i

YORK U

Scalability

- Constant $\mathcal{O}(d)$ running time for computing pairwise similarities [Lee et al – ICDM '22]
  - shown empirically and theoretically
  - d – dimension of the box embedding space
  - Constant value; set prior to model training

- Inherited by Seg2Box

- Perform such operations quickly, and can thus scale to larger datasets

Evaluation – ST2Box

Evaluating ST2Box

RQ 1) Accuracy

- How does ST2Box compare with SotA methods?

RQ 2) Spatial Experiment

- Which spatial element is best to use for expressing trajectories as sets? (sets of points, cells or road segments)

RQ 3) Robustness

- How robust is ST2Box when varying the values of the spatiotemporal weight θ?

Datasets

	T-Drive	NYC
Taxis in the urban city	Beijing	New York
# nodes	~75K	~78K
# edges	~165K	~121K
# trajectories	~348K	~634K
Observation period	1 week	1 month

- Data Preprocessing
  - Filter out trajectories with less than 10 sampling points
  - Only consider trajectories within map $\mathcal{M}$
  - Map-matching
    - Identifying the path on the road an object has taken given a sequence of GPS points

Experimental Parameters

- Spatiotemporal weight parameter $\theta = 0.5$
  - Indicates that the spatial and temporal components are both equally important
- Embedding dimensions = 128
  - Spatial embedding dimensions = 128
  - Temporal embedding dimensions = 128
- Batch size = 512
- Number of training samples
  - Positive training samples = 10
  - Negative training samples = 10
- Optimizer
  - Adam with learning rate of 0.001

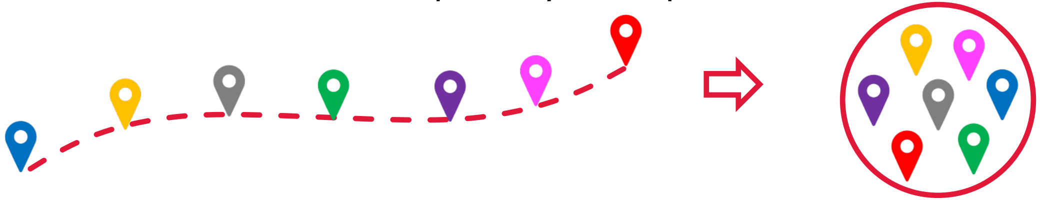
Baselines

- Deep Learning Baselines
  - CSSRNN [Xia et al, IJCAI '17] — Extension to RNN, with topological constraints
  - Traj2Vec [Yao et al, IJCNN '17] — Using sliding windows and autoencoders
  - T2Vec [Li et al, ICDE '18] — Based on Seq2Seq model
  - ST2Vec [Fang et al, KDD '22] — Fusing spatial and temporal representations (SotA)
- Box-based Architecture Baselines †
  - PTs2Box
  - Hex2Box
  - Rds2Box

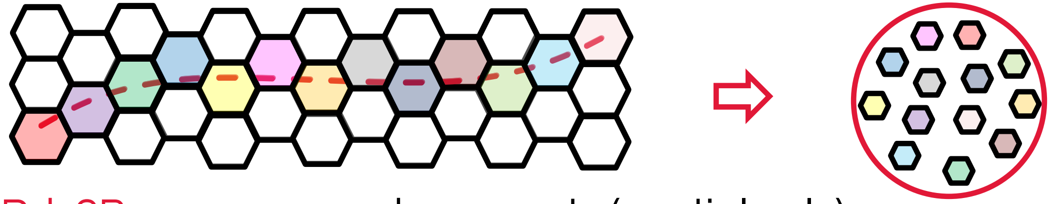
† These baselines are all ours.

Box-based Architecture Baselines

- PTs2Box – uses raw trajectory GPS points

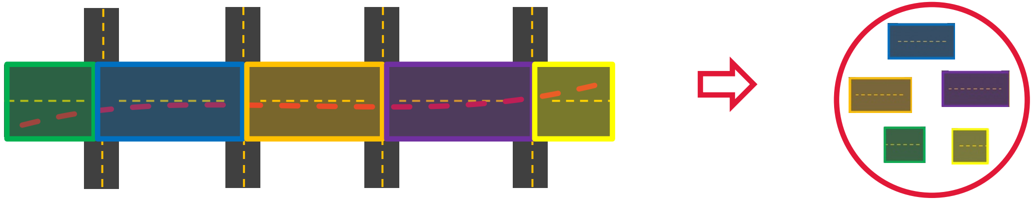


- Hex2Box – uses hexagonal-shaped block cells

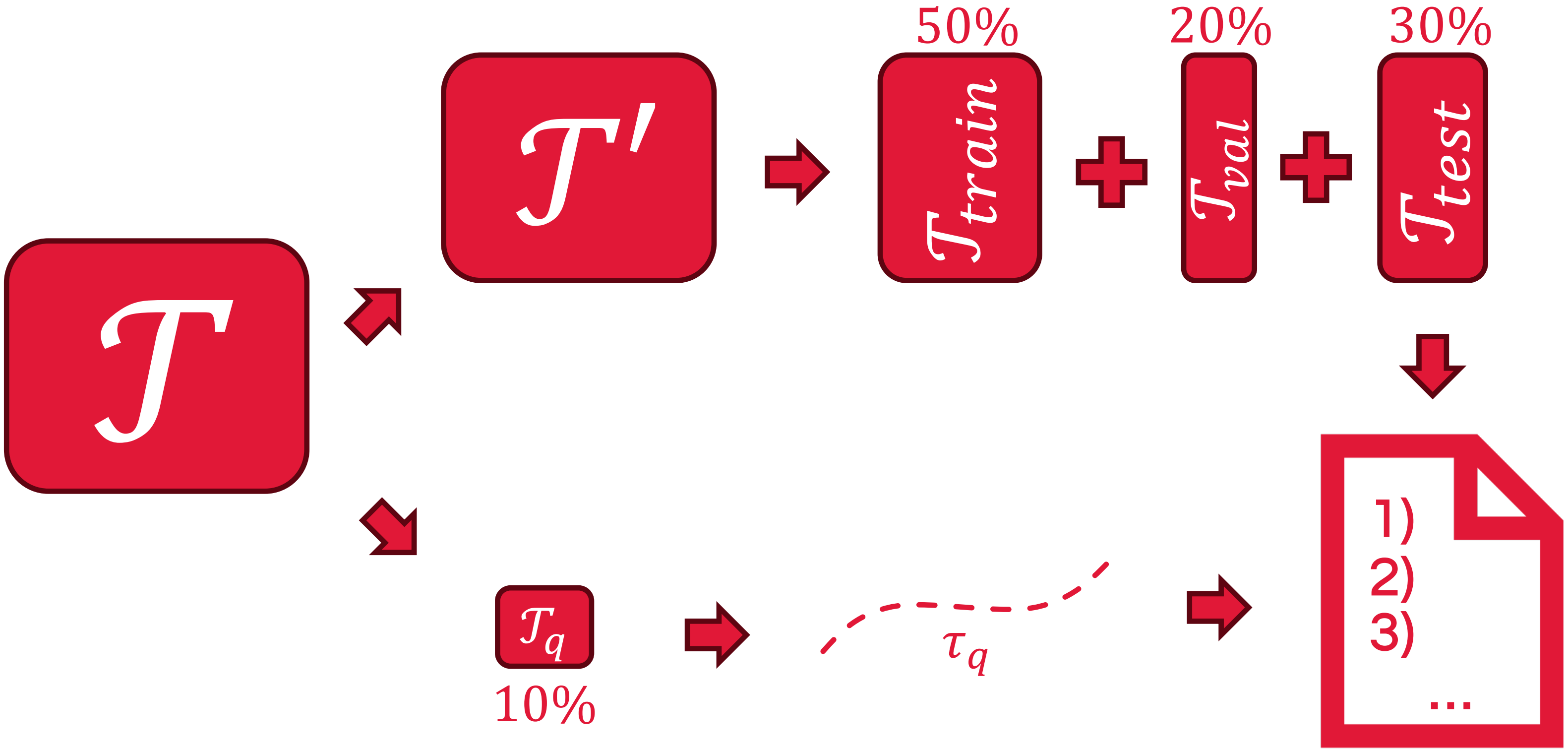


Data source: https://github.com/alifa98/point2hex

- Rds2Box – uses road segments (spatial only)



Evaluation Metrics - 1



YORK U

Evaluation Metrics - 2

- Hitting Ratio (HR) [↑]
  - The degree of overlap
- Mean Absolute Error (MAE) [↓]
  - The average of all differences
- Kendall-Tau Metric (KT) [↑]
  - The ordinal association

[↑] – Higher values indicate better model performance

[↓] – Lower values indicate better model performance

Numerical Results

	Model	TP			DITA			LCRS			NETERP			FRECHET		
		HR	MAE	KT	HR	MAE	KT	HR	MAE	KT	HR	MAE	KT	HR	MAE	KT
T-DRIVE	CSSRNN	0.618	1799	0.004	0.714	1667	0.228	0.642	2558	0.189	0.669	1746	0.107	0.705	1738	0.004
	TRAJ2VEC	0.651	1774	0.037	0.695	1642	0.272	0.636	2570	0.168	0.683	1732	0.111	0.679	1719	0.110
	T2VEC	0.640	1783	0.010	0.730	1650	0.259	0.629	2564	0.182	0.654	1760	0.046	0.728	1723	0.009
	ST2VEC	<u>0.745</u>	<u>1705</u>	<u>0.232</u>	<u>0.822</u>	<u>1565</u>	<u>0.413</u>	<u>0.781</u>	<u>2488</u>	<u>0.313</u>	<u>0.760</u>	<u>1683</u>	<u>0.210</u>	<u>0.826</u>	<u>1656</u>	<u>0.244</u>
	PTS2BOX	0.679	1766	0.096	0.782	1624	0.320	0.656	2551	0.203	0.698	1719	0.128	0.753	1702	0.131
	HEX2BOX	0.709	1744	0.157	0.803	1609	0.350	0.687	2548	0.245	0.711	1704	0.167	0.792	1689	0.165
	RDS2BOX	0.718	1729	0.201	0.810	1598	0.387	0.722	2534	0.283	0.749	1697	0.195	0.813	1661	0.207
	ST2BOX	**0.823**	**1682**	**0.319**	**0.862**	**1556**	**0.466**	**0.814**	**2465**	**0.350**	**0.786**	**1648**	**0.231**	**0.842**	**1635**	**0.269**
	% Impr.	10.5%	1.4%	37.5%	4.9%	0.6%	12.8%	4.2%	0.9%	11.8%	3.4%	2.1%	10.0%	1.9%	1.3%	10.2%
NYC	CSSRNN	0.649	2619	0.103	0.672	2491	0.027	0.621	2207	0.086	0.616	2069	0.101	0.679	1673	0.144
	TRAJ2VEC	0.611	2608	0.114	0.669	2470	0.039	0.659	2179	0.118	0.649	2090	0.089	0.720	1640	0.171
	T2VEC	0.678	2628	0.097	0.665	2453	0.060	0.640	2192	0.105	0.686	2041	0.122	0.707	1602	0.197
	ST2VEC	<u>0.842</u>	<u>2550</u>	<u>0.274</u>	<u>0.827</u>	<u>2348</u>	<u>0.207</u>	<u>0.749</u>	<u>2117</u>	<u>0.230</u>	<u>0.801</u>	<u>1936</u>	<u>0.243</u>	<u>0.820</u>	<u>1492</u>	<u>0.348</u>
	PTS2BOX	0.725	2602	0.120	0.690	2417	0.108	0.682	2165	0.136	0.716	2009	0.166	0.748	1578	0.253
	HEX2BOX	0.751	2584	0.198	0.741	2401	0.139	0.703	2146	0.188	0.741	1986	0.183	0.775	1561	0.286
	RDS2BOX	0.760	2561	0.225	0.784	2367	0.174	0.716	2130	0.202	0.763	1958	0.207	0.791	1539	0.304
	ST2BOX	**0.879**	**2498**	**0.333**	**0.883**	**2319**	**0.265**	**0.817**	**2085**	**0.283**	**0.835**	**1893**	**0.272**	**0.893**	**1463**	**0.399**
	% Impr.	4.4%	2.0%	21.5%	6.8%	1.2%	28.0%	9.1%	1.5%	23.0%	4.2%	2.2%	11.9%	8.9%	1.9%	14.7%

Table 5.2: Numerical results for similarity search task; bold/underlined numbers indicate best/second best method respectively. The last row shows the % improvement attained by ST2BOX from the state-of-the-art ST2VEC.

RQ 1) Accuracy Performance – Key Observations

- ☑ ST2Vec and ST2Box consistently outperform the baselines
- ☑ ST2Box consistently outperform ST2Vec
- ☑ ST2Box learns more refined levels of spatiotemporally-enriched information

RQ 2) Spatial Experiment - Results

PTs2Box does not do well

- Noisy, unmap-matched data

Hex2Box is also outperformed

- Under/overestimate the exact paths

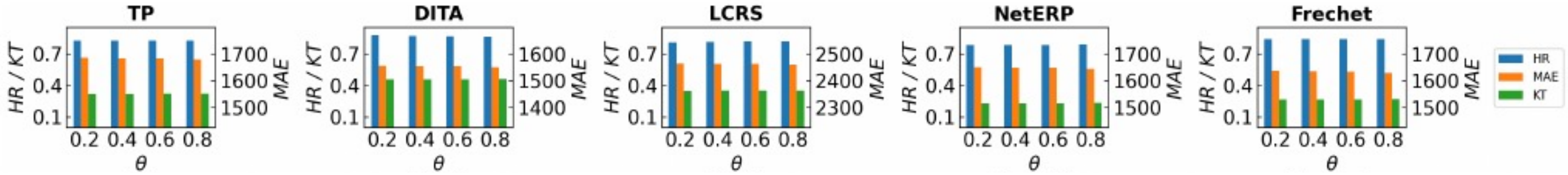
Rds2Box yields the best performance

- Improved by ST2Box – capture the temporal aspect

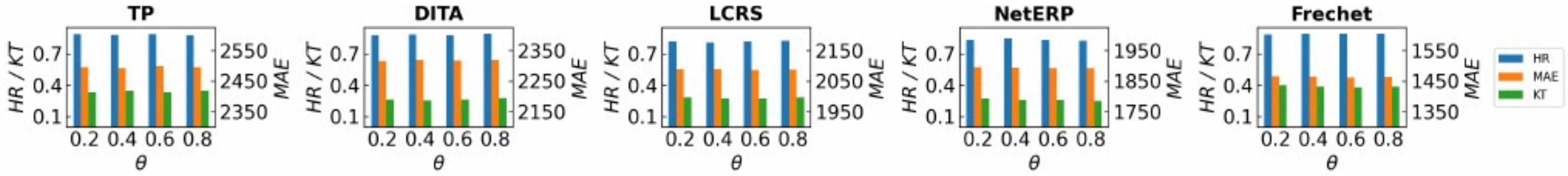
RQ 3) Robustness Study

- Varying the spatiotemporal weight $\theta \in [0.0, 0.1, \dots, 1.0]$
- Performance of ST2Box remains stable across all three evaluation metrics for both datasets
- Provides support for a variety of the applications

T-Drive Dataset



NYC Dataset



YORK U

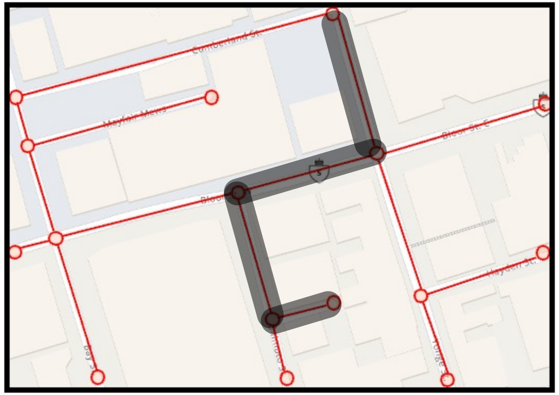
Constructing Pathlet Dictionaries

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]

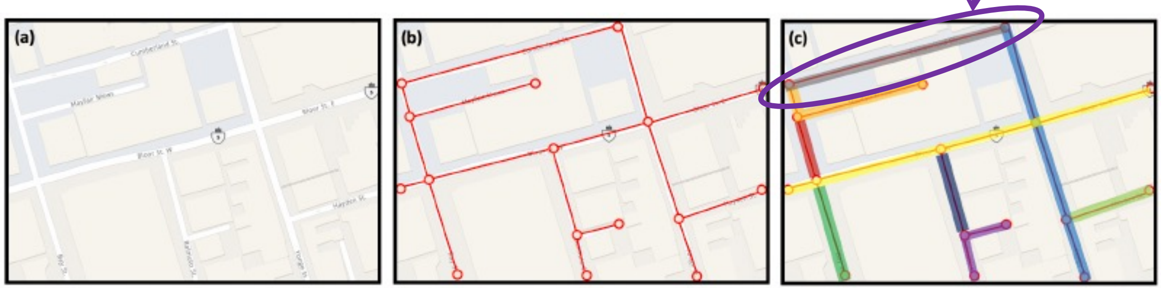
  - Pathlet
  - Subtrajectory
  - Trajectory Segments
  - Fragments
  - ...



YORK U

Brief Background: Pathlets

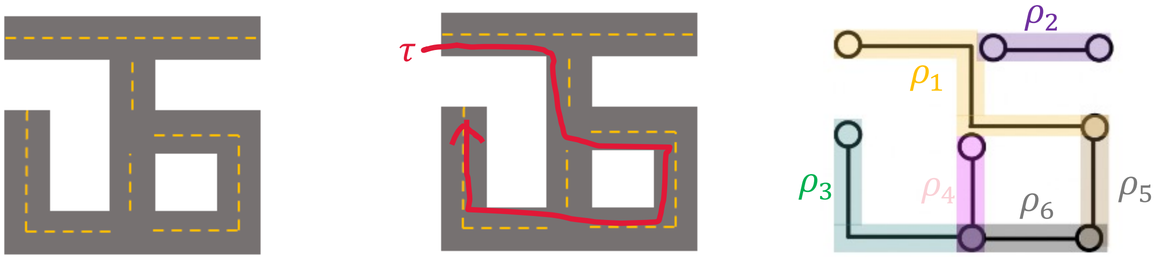
Grey pathlet has two neighbors:
orange and blue pathlets



- 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

Pathlet-based Representation of a Trajectory

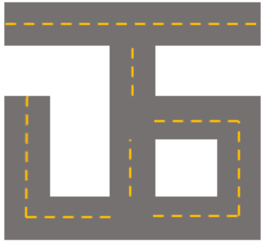
Denoted by $\Phi(\tau) = \{\rho^{(1)}, \rho^{(2)}, \dots, \rho^{(k)}\}$



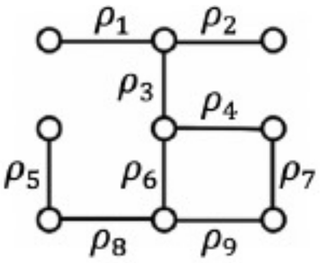
(a) (b) (c)

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

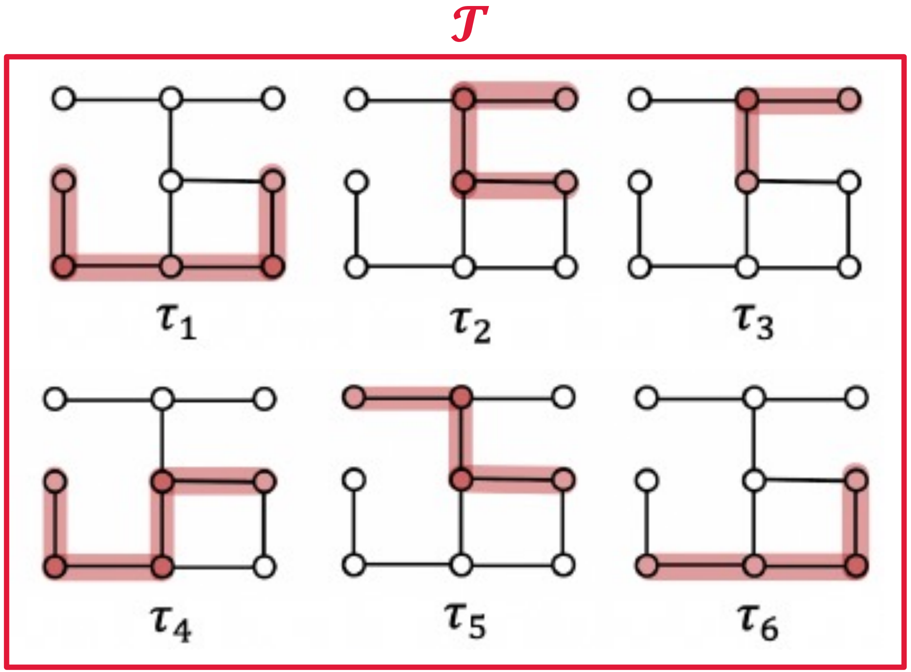
Trajectory Traversal Set



(a)



(b)



- Denoted by

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

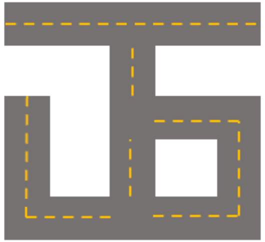
- Pathlet Weights – importance in the road network

$$\Lambda(\rho_1) = \{\tau_5\} \qquad \Lambda(\rho_4) = \{\tau_2, \tau_4, \tau_5\} \qquad \Lambda(\rho_7) = \{\tau_1, \tau_6\}$$

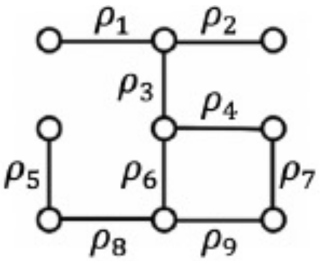
$$\Lambda(\rho_2) = \{\tau_2, \tau_3\} \qquad \Lambda(\rho_5) = \{\tau_1, \tau_4\} \qquad \Lambda(\rho_8) = \{\tau_1, \tau_4, \tau_6\}$$

$$\Lambda(\rho_3) = \{\tau_2, \tau_3, \tau_5\} \qquad \Lambda(\rho_6) = \{\tau_4\} \qquad \Lambda(\rho_9) = \{\tau_1, \tau_6\}$$

Pathlet Dictionary



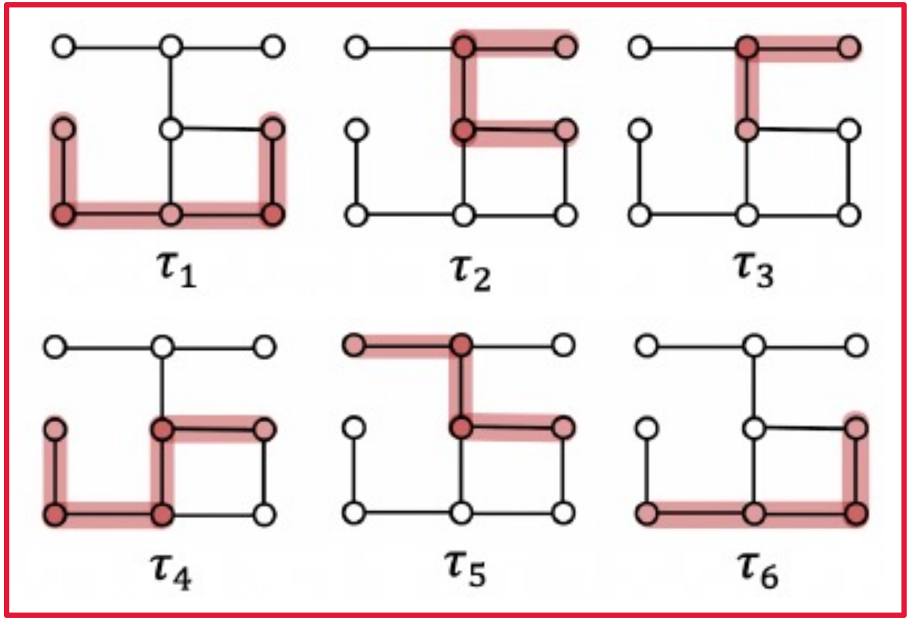
(a)



(b)

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	$\{\tau_1, \tau_6\}$

$\mathcal{T}$



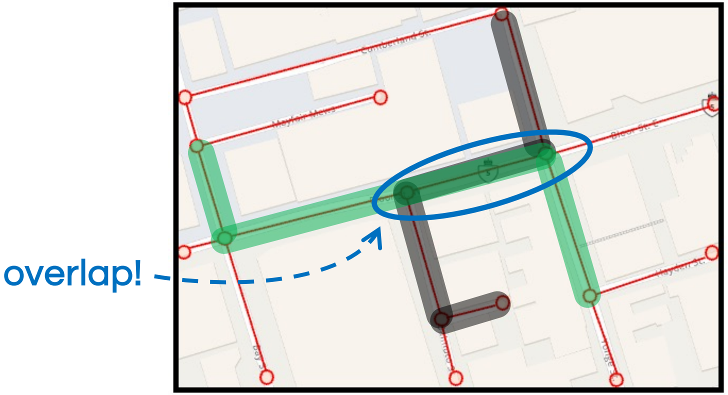
Existing Works

Existing Works and Limitations

- Existing works

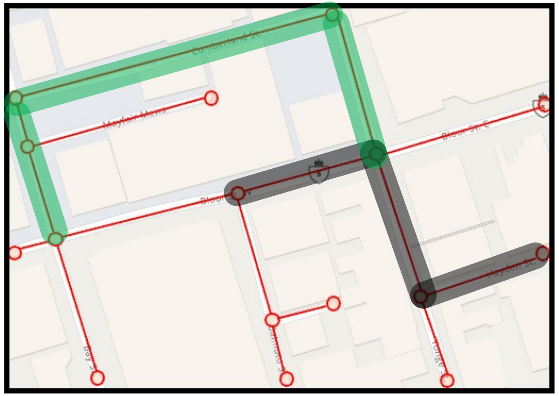
  [Panagiotakis et al – TKDE '12, Chen et al – SIGSPATIAL '13, Sankararaman et al – SIGSPATIAL '13, Agarwal et al – PODS '18, Li et al – TSAS '18, Zhao et al – CIKM '18]

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



Overlapping Pathlets

(Top-down Approach)



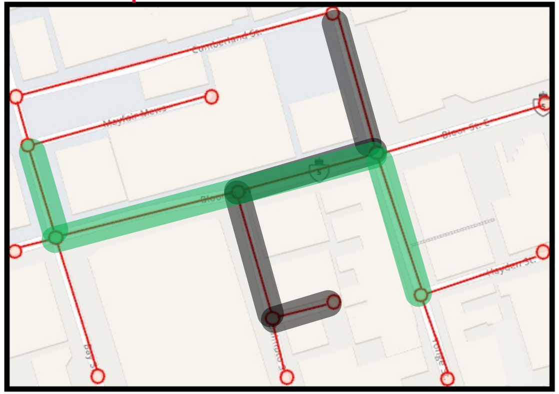
Edge-disjoint Pathlets

(Bottom-up Approach)

YORK U

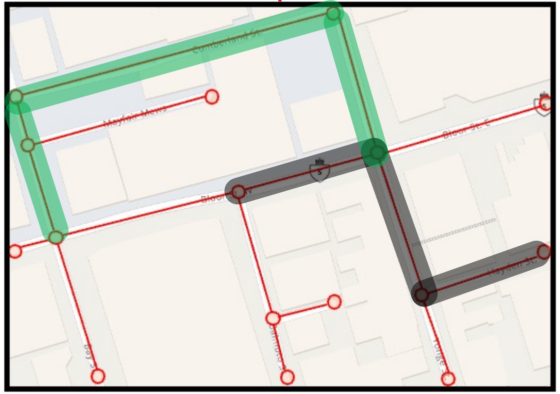
Top-down vs Bottom-up Methods

Top-down Methods



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

Bottom-up Methods



- 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

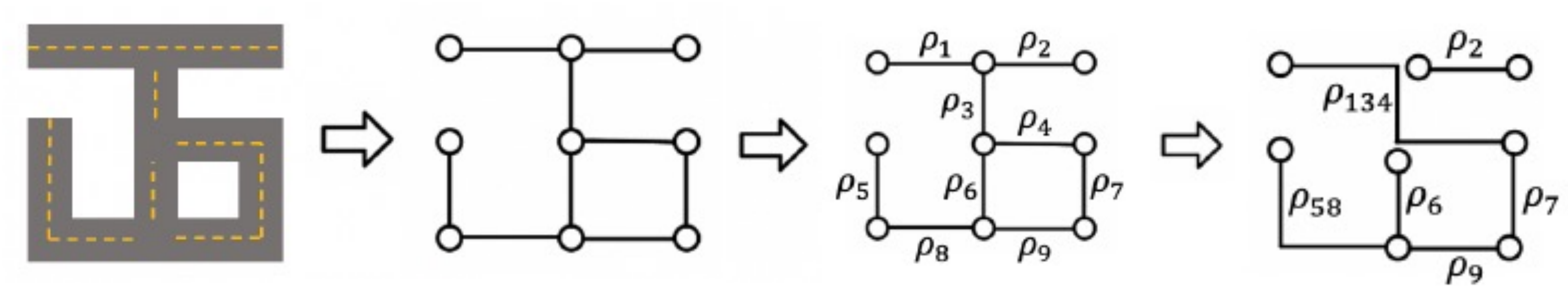
the number of road segments

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%

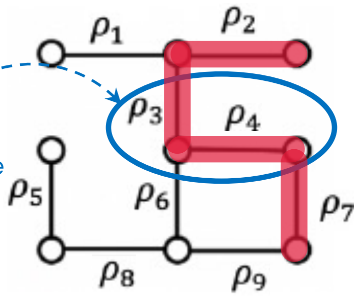
YORK U

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

$\{\rho_2, \rho_7\}$

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

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

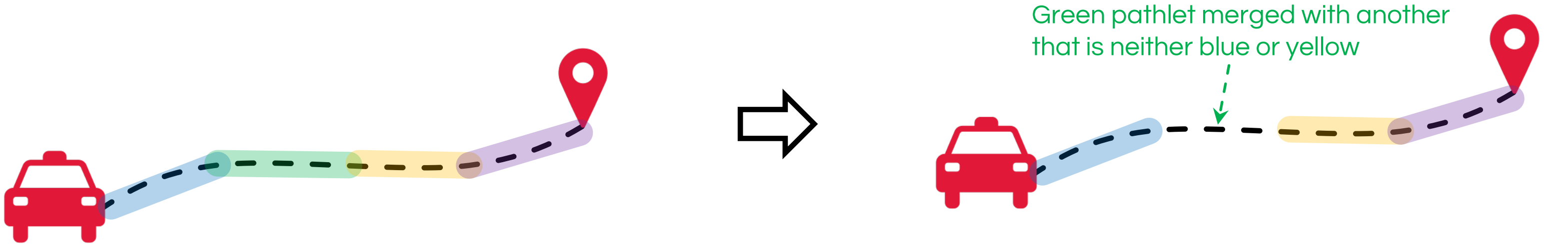
Notice that μ is monotonically non-increasing at each step of the iteration

Trajectory is lost/discarded once μ reaches zero!

YORK U

Remark

- Ignoring trajectory representability?
  - A portion of a trajectory that cannot be represented by the pathlets is considered lost
  - Degrades performance in constructing good quality dictionaries



Initial State

	Representability	Trajectory Lost?
With Representability	100%	No
Without Representability	N/A	No

After the Merge

	Representability	Trajectory Lost?
With Representability	75%	No
Without Representability	N/A	Yes

PD 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, $\bar{\mu}$

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

YORK U

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

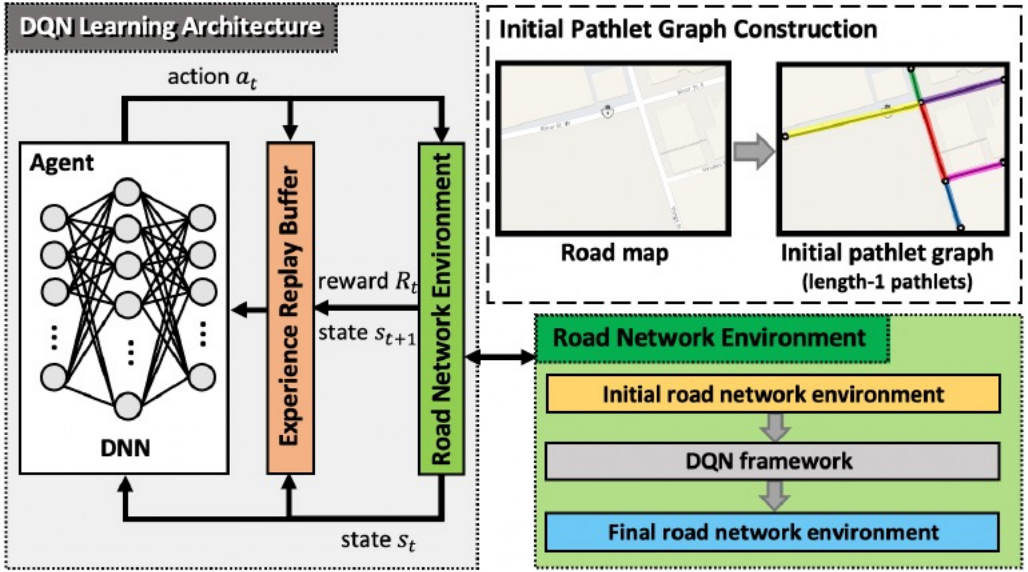
    Trajectory loss constraint $L_{traj} < M$

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

Methodology - PathletRL

PathletRL - Overview

- Extracting candidate pathlets
- Deep Reinforcement Learning framework



Extracting Candidate Pathlets - Pseudocode

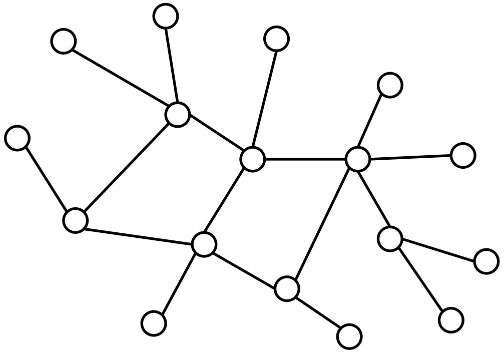
Algorithm 4.2: Candidate Pathlet Set Extraction Algorithm

Input : The road network $\mathcal{G}\langle\mathcal{V},\mathcal{E}\rangle$, the trajectory set $\mathcal{T}$, integer χ, the maximum trajectory loss M and the average trajectory representability threshold $\hat{\mu}$.

Output : The χ-order candidate pathlet set $\mathbb{S}$ of merged pathlets with a trajectory loss not exceeding M

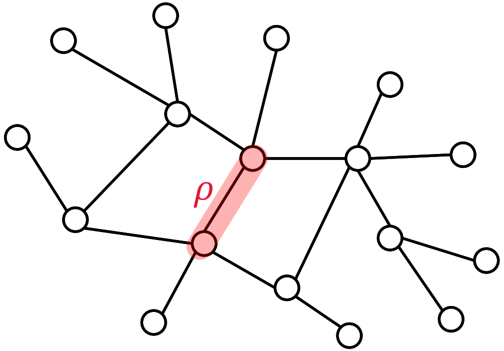
`/* Initialization */`
$\mathcal{G}_p\langle\mathcal{V}_p,\mathcal{E}_p\rangle \leftarrow \textsc{ExtractPathletGraph}(\mathcal{G}\langle\mathcal{V},\mathcal{E}\rangle)$
$\ell \leftarrow 1$ `// Size of the initial length 1 pathlets`
$\mathcal{T}^* \leftarrow \mathcal{T}$; $\mathbb{S} \leftarrow \varnothing$
$\phi \leftarrow \frac{1}{|\mathcal{T}^*|}\sum_{\tau\in\mathcal{T}^*}|\Phi(\tau)|$; $\bar{\mu} \leftarrow \frac{1}{|\mathcal{T}^*|}\sum_{\tau\in\mathcal{T}^*}\mu(\tau)$
`/* Setup traj loss and utility dictionaries */`
$T_D \leftarrow \textsc{Dict}()$; $U_D \leftarrow \textsc{Dict}()$
$\rho \leftarrow \textsc{Rand}(\mathcal{E}_p)$ `// Uniformly pick` $\rho\in\mathcal{E}_p$ `at random`
`/* Repeat until all pathlets are processed */`
`/* Or when traj loss exceeds the maximum */`
while $\mathcal{E}_p \neq \mathbb{S}$ ***or*** $\textit{sum}(T_D.\textsc{Values}()) < M$ ***or*** $\bar{\mu} \geq \hat{\mu}$ **do**
    `/* Initially set` $\mathcal{G}_p$`'s utility associated to the curr pathlet` ρ `to be 0 */`
    $U_D[\rho] = 0$
    `/* For each of` ρ`'s unprocessed neighbors */`
    foreach $\hat{\rho}$ ***in*** $\Psi(\rho)\setminus\mathbb{S}$ **do**
        $U_D[\hat{\rho}] \leftarrow \textsc{ComputeUtil}(\textsc{Merge}(\rho,\hat{\rho}))$
        $T_D[\hat{\rho}] \leftarrow \textsc{GetAllTrajLost}\ (\textsc{Merge}(\rho,\hat{\rho}),\mathcal{T}^*)$
    `/* Find the one with the highest utility */`
    $\rho^* \leftarrow \operatorname{argmax}_{\rho\in key} U_D[key]$
    if $\rho^* = \rho$ ***or*** $\ell > \chi$ **then** `// Merge not necessary`
        $\mathbb{S} \leftarrow \mathbb{S}\cup\{\rho\}$
        $\rho \leftarrow \textsc{Rand}(\mathcal{E}_p\setminus\mathbb{S})$ `// Pick new pathlet`
        if $\rho = \varnothing$ **then** `// All pathlets processed`
            break
        $\ell \leftarrow 1$ `// Reset pathlet length`
    else `// Merge recommended`
        $\rho_{merged} \leftarrow \textsc{Merge}(\rho,\hat{\rho})$
        $\mathcal{E}_p \leftarrow (\mathcal{E}_p\setminus\{\rho,\hat{\rho}\})\cup\{\rho_{merged}\}$
        $\mathcal{T}^* \leftarrow \mathcal{T}^*\setminus T_D[\rho^*]$
        $\phi \leftarrow \frac{1}{|\mathcal{T}^*|}\sum_{\tau\in\mathcal{T}^*}|\Phi(\tau)|$; $\bar{\mu} \leftarrow \frac{1}{|\mathcal{T}^*|}\sum_{\tau\in\mathcal{T}^*}\mu(\tau)$
        $\rho \leftarrow \rho_{merged}$; $\ell \leftarrow \ell + 1$
return $\mathbb{S}$

Extracting Candidate Pathlets - Example



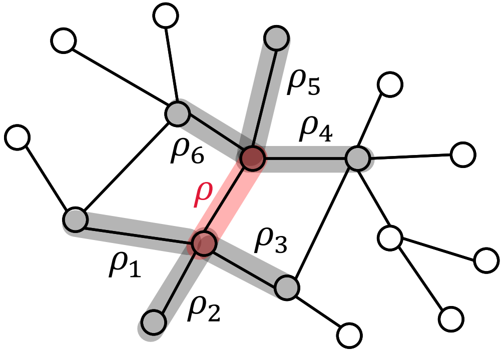
Initialize length-1 pathlets

Extracting Candidate Pathlets - Example



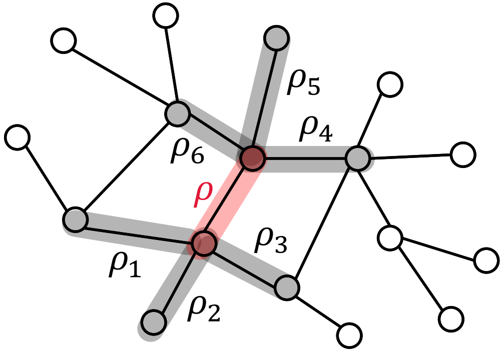
Choose pathlet ρ uniformly at random

Extracting Candidate Pathlets - Example



Identify all neighbors of ρ

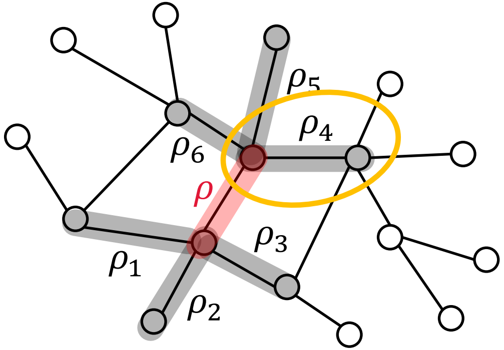
Extracting Candidate Pathlets - Example



Compute utility of ρ_{merge}

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

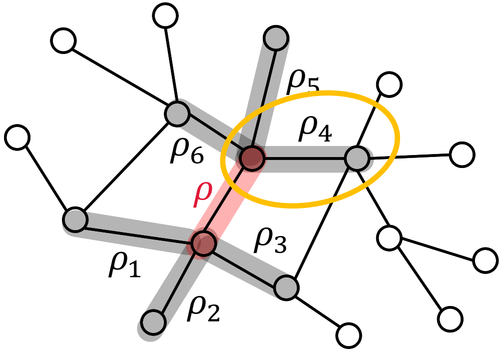
Extracting Candidate Pathlets - Example



Obtain ρ^* with the highest utility

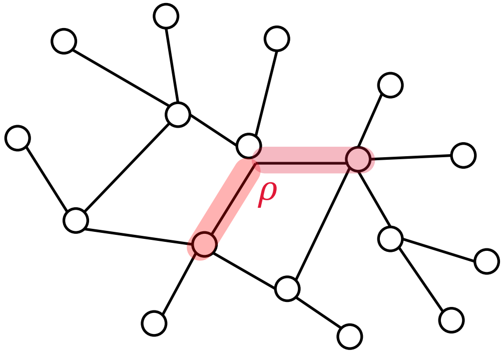
ρ_{merge}	Utility
MERGED(ρ, ρ_1)	+0.7
MERGED(ρ, ρ_2)	+1.8
MERGED(ρ, ρ_3)	-1.6
MERGED(ρ, ρ_4)	+5.5
MERGED(ρ, ρ_5)	-3.2
MERGED(ρ, ρ_6)	+2.9

Extracting Candidate Pathlets - Example



Merge ρ and ρ_4

Extracting Candidate Pathlets - Example



New current pathlet ρ

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

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

- 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

Experimental Parameters

- Deep neural network
  - Consists of FC layers of 128, 64 and 32 hidden neurons
  - ReLU activation function
  - Adam optimizer with learning rate of 0.001
  - Dropout = 0.2
- DQN parameters
  - A total of 5 episodes for each of the 100 iterations
  - Size of the experience replay buffer is 100,000
  - Memory minibatch size is 64
  - Discount factor of $\gamma = 0.99$
- χ-order candidate set: $\chi = 10$
- Maximum trajectory loss is 25% and average representability threshold is 80%
- Objective parameter $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 0.25$

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:

- All these metrics indicate that lower values are better, except the last one that indicates that 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

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])

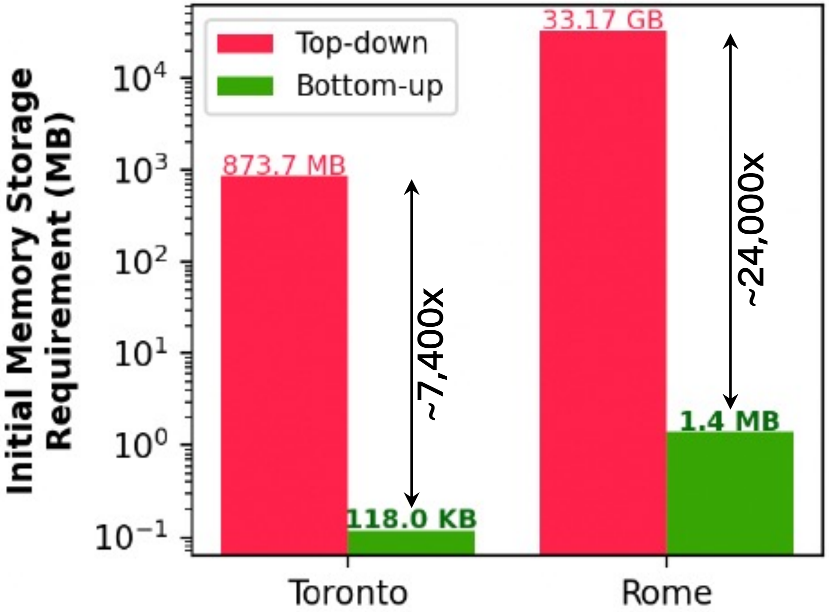
[1] Agarwal et al, PODS '18

[26] Chen et al, SIGSPATIAL '13

YORK U

RQ 2) Memory Efficiency

Bottom-up approaches outperform top-down methods

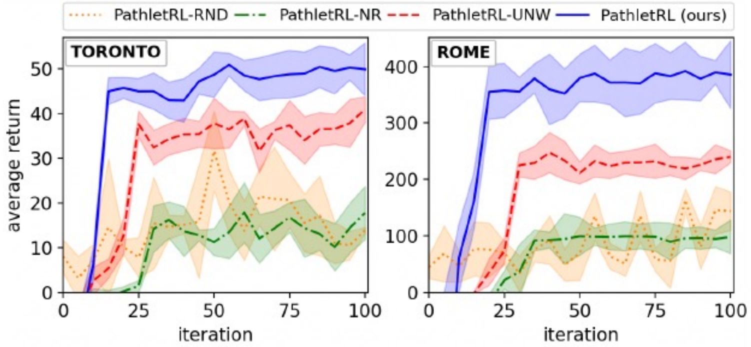


YORK U

RQ 3) Ablation Study – Average Returns

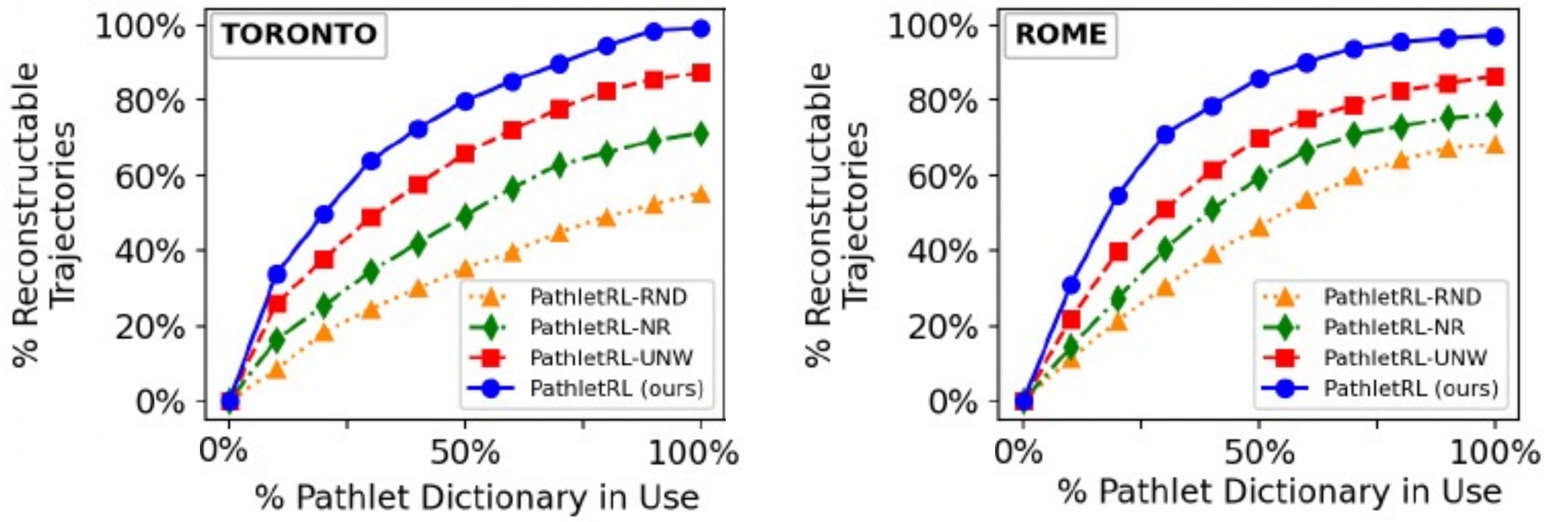
PathletRL Algorithm	Representability Measure	Weighted Networks	Deep Learning Policy
PathletRL-NR	✗	✓	✓
PathletRL-RND	✓	✓	✗
PathletRL-UNW	✓	✗	✓
PathletRL (ours)	✓	✓	✓

- 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



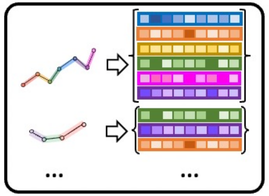
YORK U

RQ 4) Partial Trajectory Reconstruction

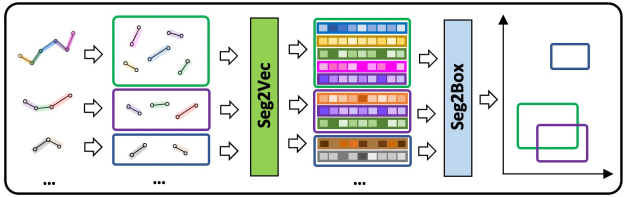


Conclusions

Summary and Contributions



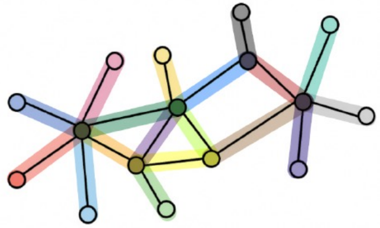
Trajectories as Sets



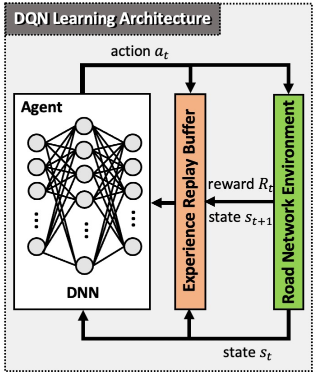
ST2Box: Seg2Vec and Seg2Box

ST2Box Properties

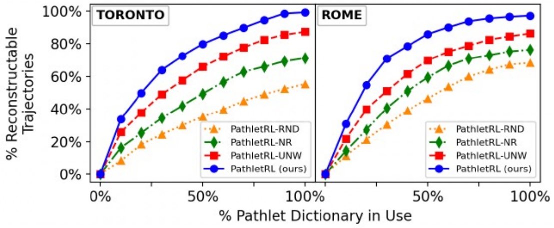
- Accurate
- Versatile
- Generalizable
- Robust
- Fast and Scalable



Edge-disjoint pathlets



Deep Reinforcement Learning (DQN)



Partial trajectory reconstruction ~85%

Papers Published/Under Review

- G. Alix and M. Papagelis, "ST2Box: Trajectory Similarity Learning using Set2Box Representations," IEEE ICDM '23 (under review).
- G. Alix and M. Papagelis, "PathletRL: Trajectory Pathlet Dictionary Construction using Reinforcement Learning," ACM SIGSPATIAL '23 (under review).
- A. Faraji, J. Li, G. Alix, M. Alsaeed, N. Yanin, A. Nadiri, and M. Papagelis, "Point2Hex: Higher-order Mobility Flow Data and Resources," ACM SIGSPATIAL '23 (under review).
- T. Pechlivanoglou, G. Alix, N. Yanin, J. Li, F. Heidari, and M. Papagelis, "Microscopic modeling of spatiotemporal epidemic dynamics," ACM SIGSPATIAL '22 (SpatialEpi '22 workshop), https://doi.org/10.1145/3557995.3566116.
- 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," IEEE MDM '22, https://doi.org/10.1109/MDM55031.2022.00063.

Thank you!

Questions?