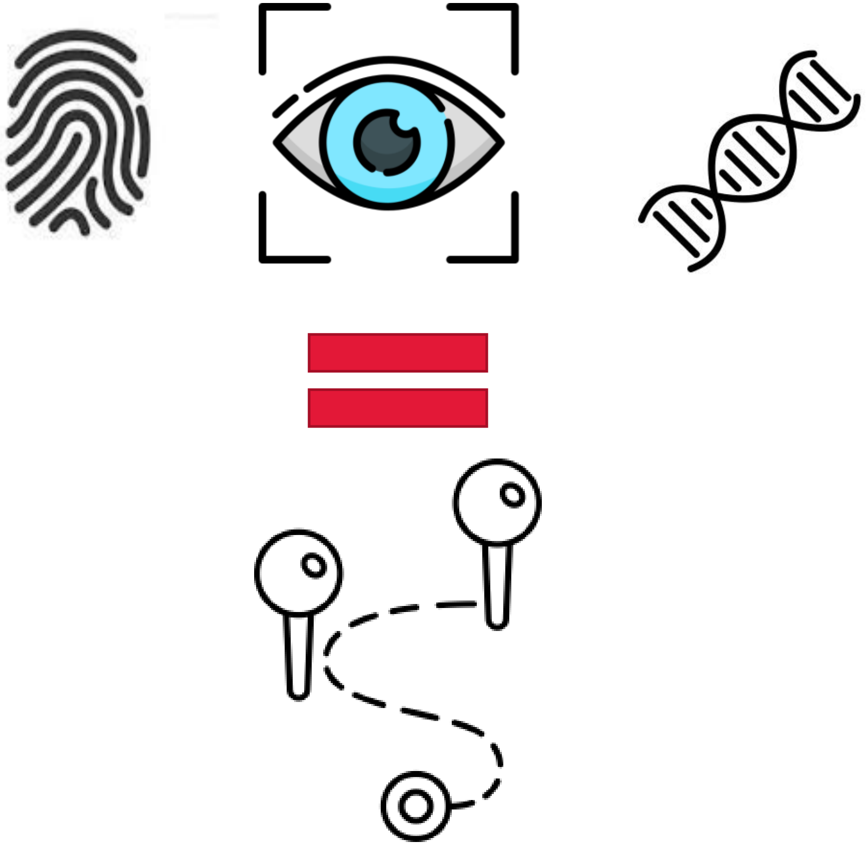
Trajectory-User Linking using Higher-order Mobility Flow Representations

Supervisor: Manos Papagelis

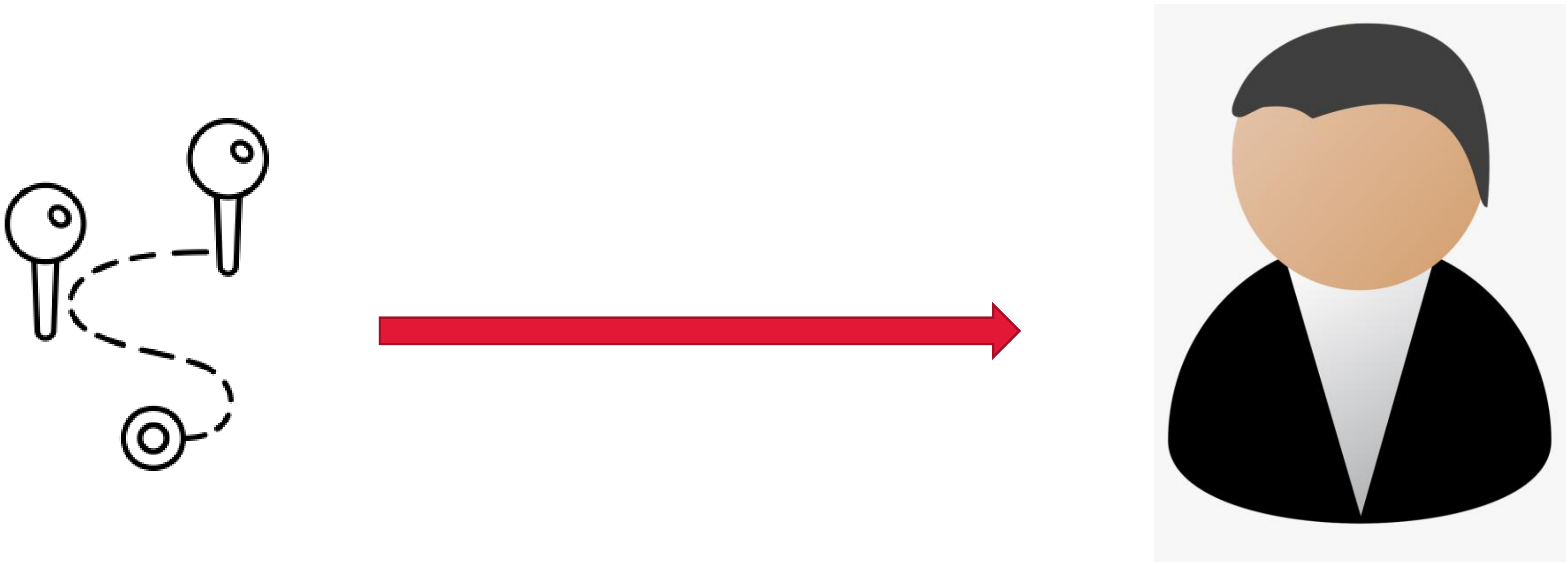
M.SC. THESIS OF MAHMOUD ALSAEED

LASSONDE SCHOOL OF ENGINEERING | YORK U



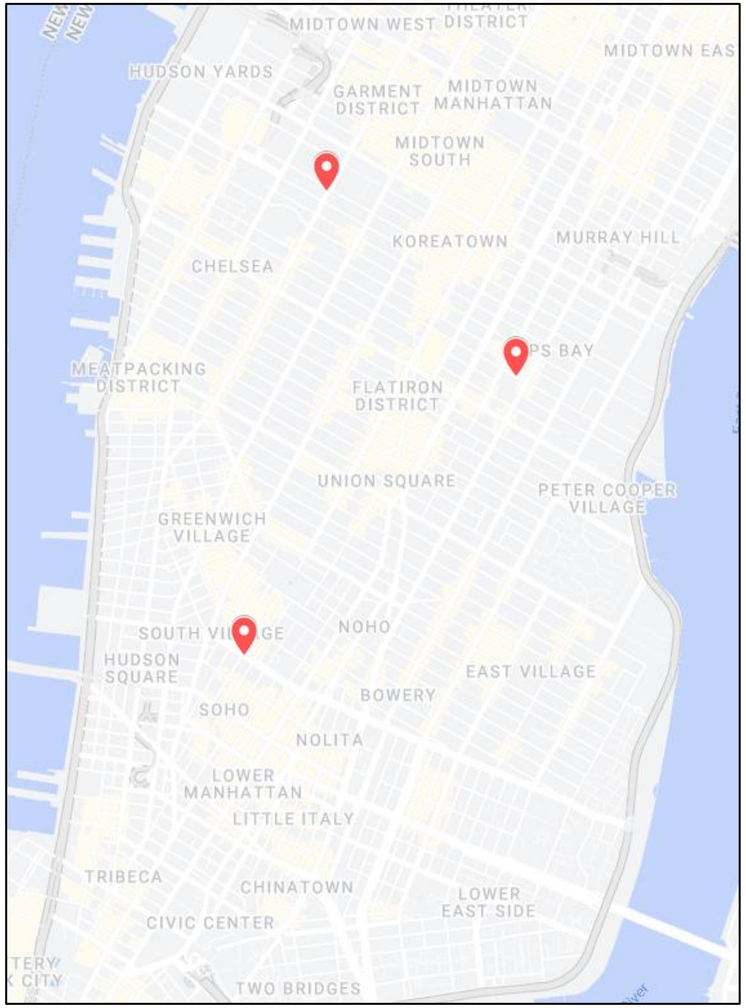
trajectories are
unique to each person

Trajectory-user Linking (TUL)



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

Data for Trajectory-user Linking (TUL)



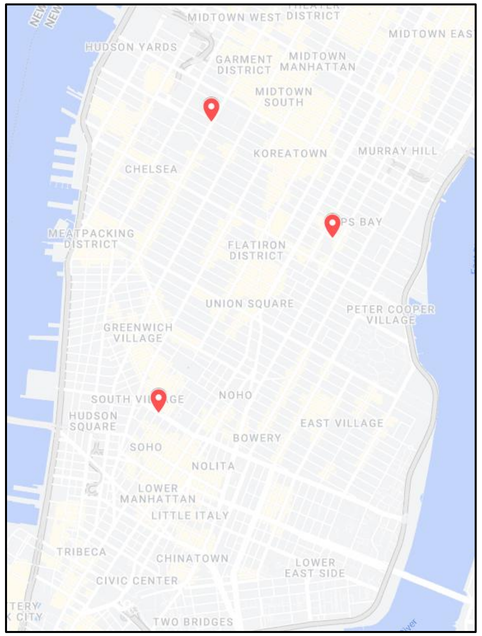
Check-ins Trajectory

Mobility flow

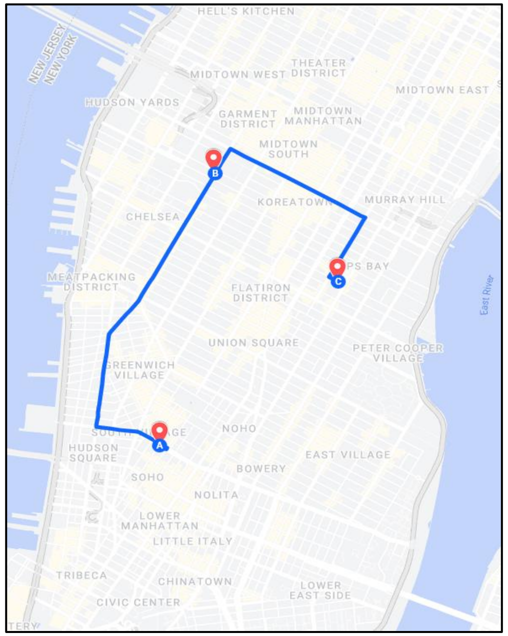
Why do the limitations exist ?

- Check-ins data

- Active data collection



Check-ins trajectory



Mobility flow

Limitations of the current approaches

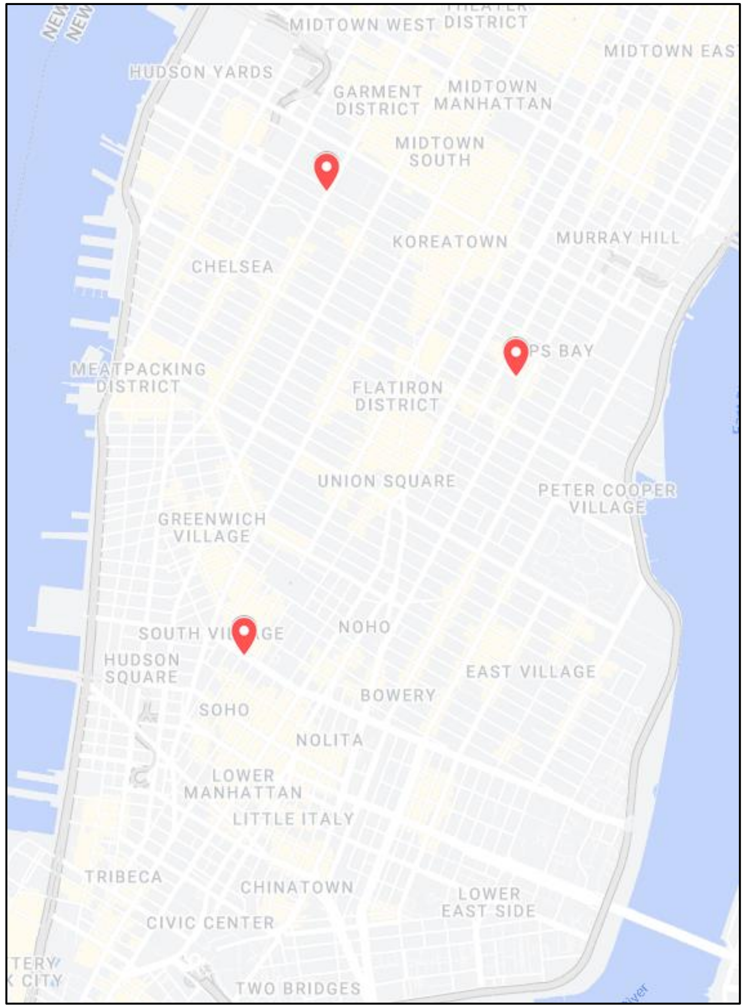
- Data Quality
  - low accuracy and completeness
- Data sparsity
  - limited data
- Imbalanced Data
  - 80% of the data is generated by 20% of the users

Problem Definition

What is a check-in trajectory ?

- Check-in record/visit
  - $r = (u, p, t, \langle x, y \rangle)$

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



Check-ins Trajectory

Problem Definition

- Trajectory-user linking
  - aims to link anonymous trajectories to users who generate them

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

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

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \boxed{\mathcal{L}(y_i - y_i')}$$

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

$\mathcal{L}(\cdot)$ *is the loss between the predicted label* y_i' *and the true label* y_i

LASSONDE SCHOOL OF ENGINEERING | YORK U

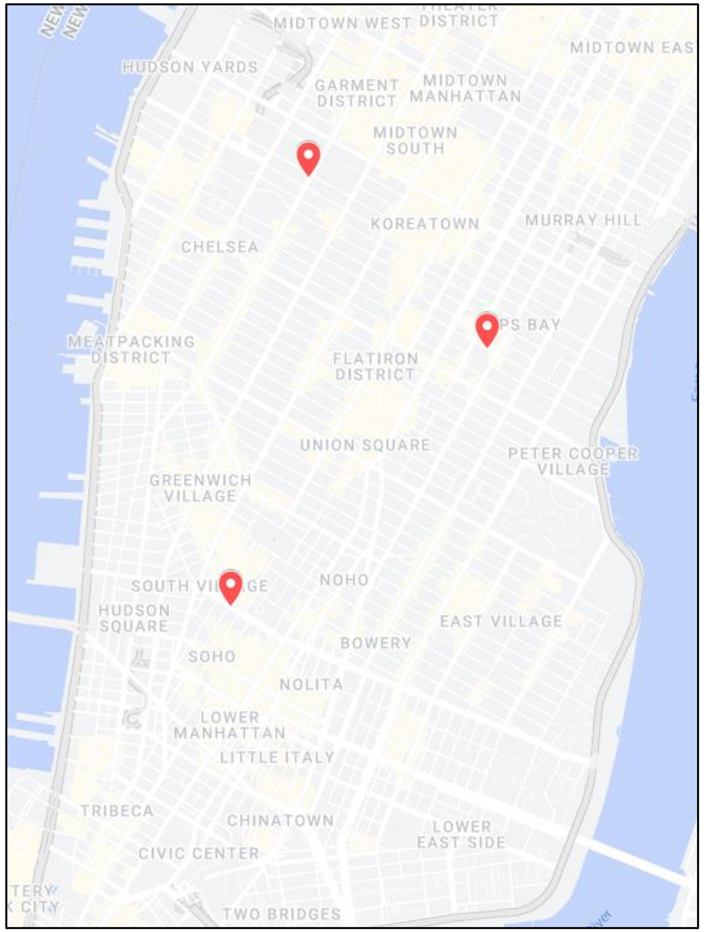
Methodology

Overview

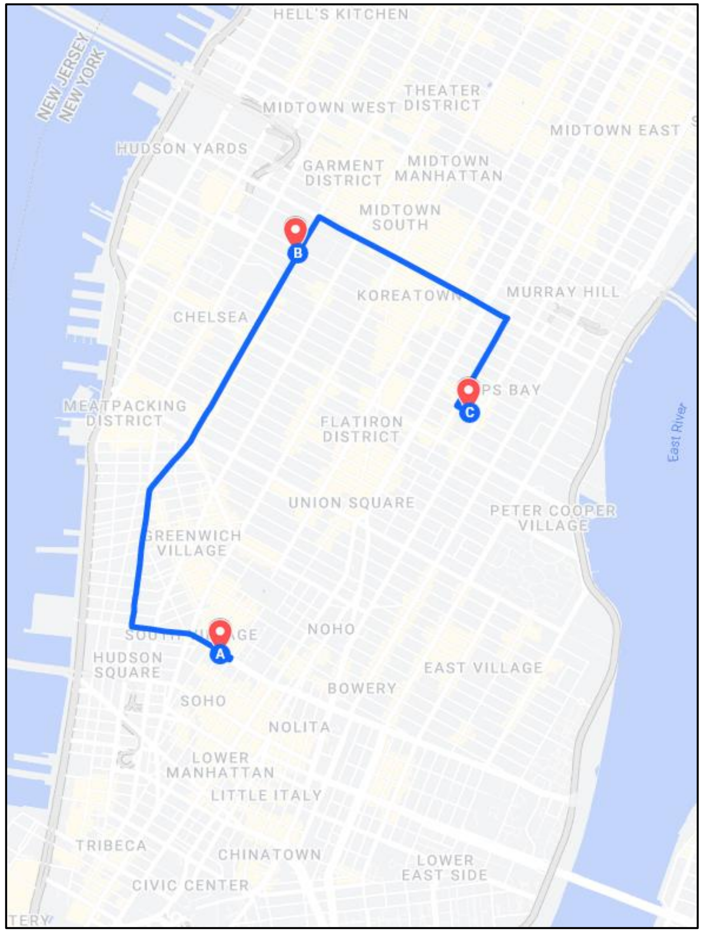
- Generating higher-order mobility flow representations
  - generating mobility flow data from check-ins
  - generating higher-order mobility flow and check-ins

- Modeling trajectory-User linking

Generating Mobility flow data

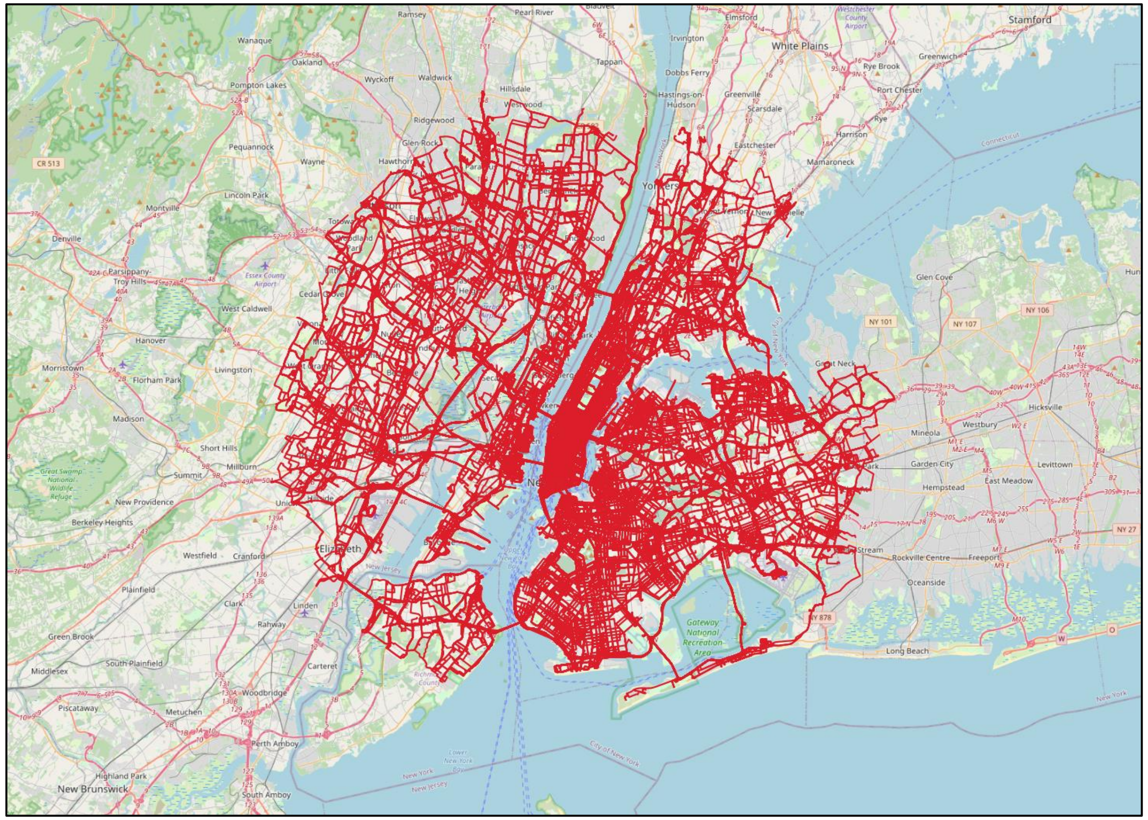


Check-ins Trajectory

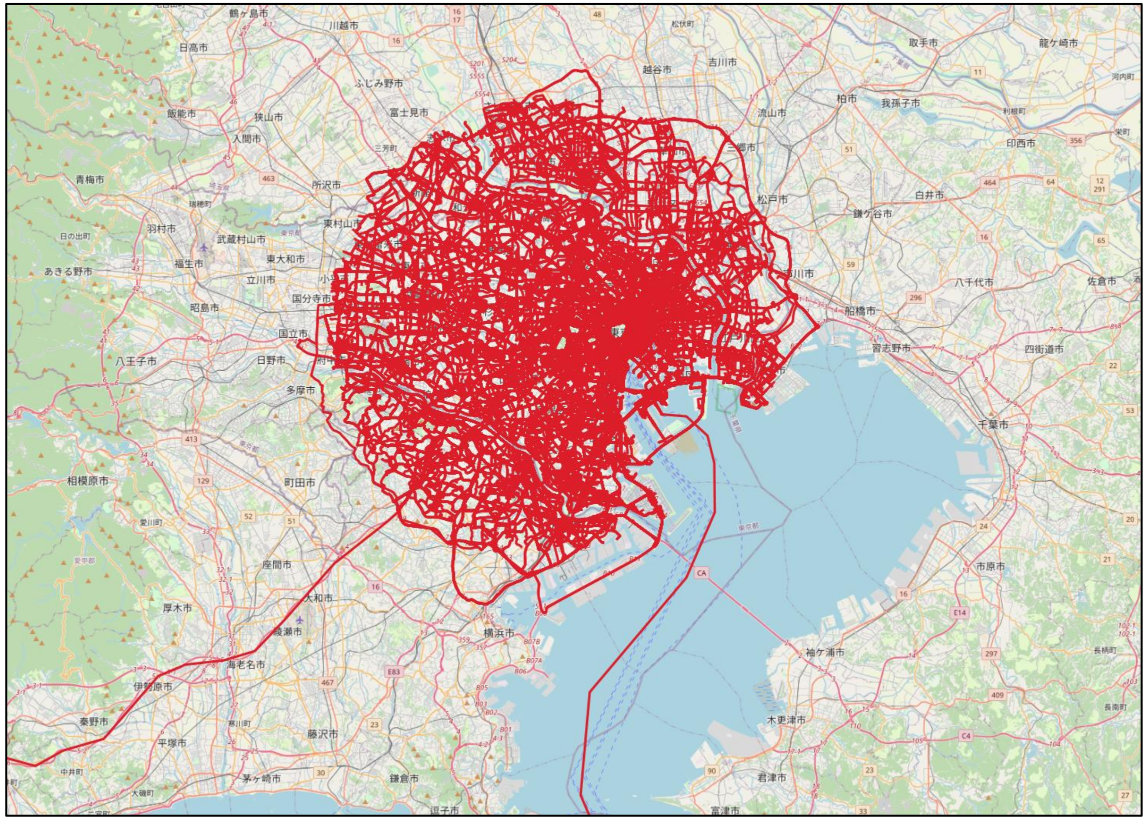


Mobility flow

Mobility flow of NYC and TKY

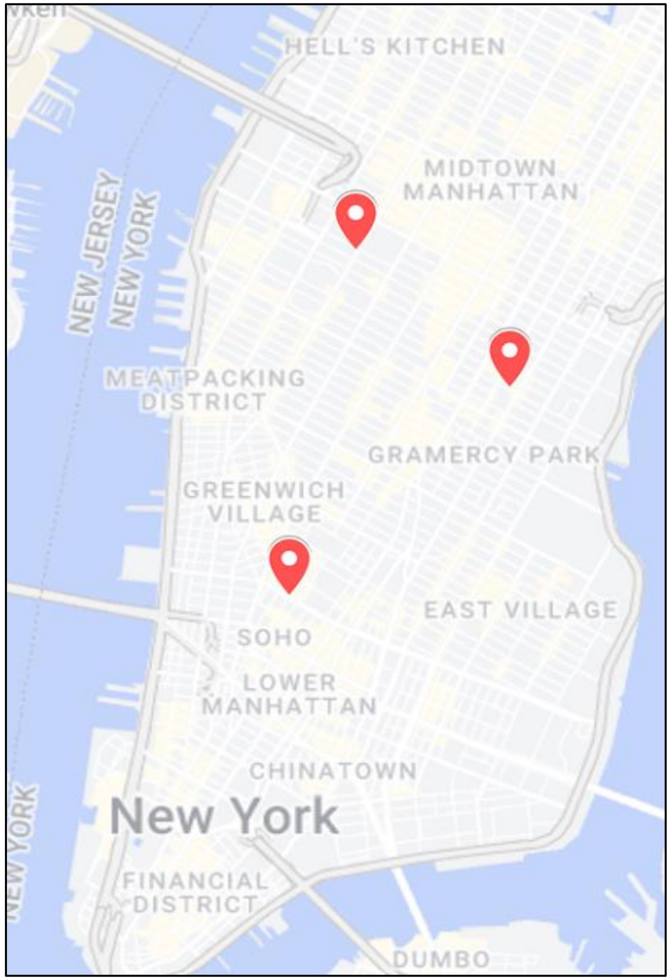


NYC

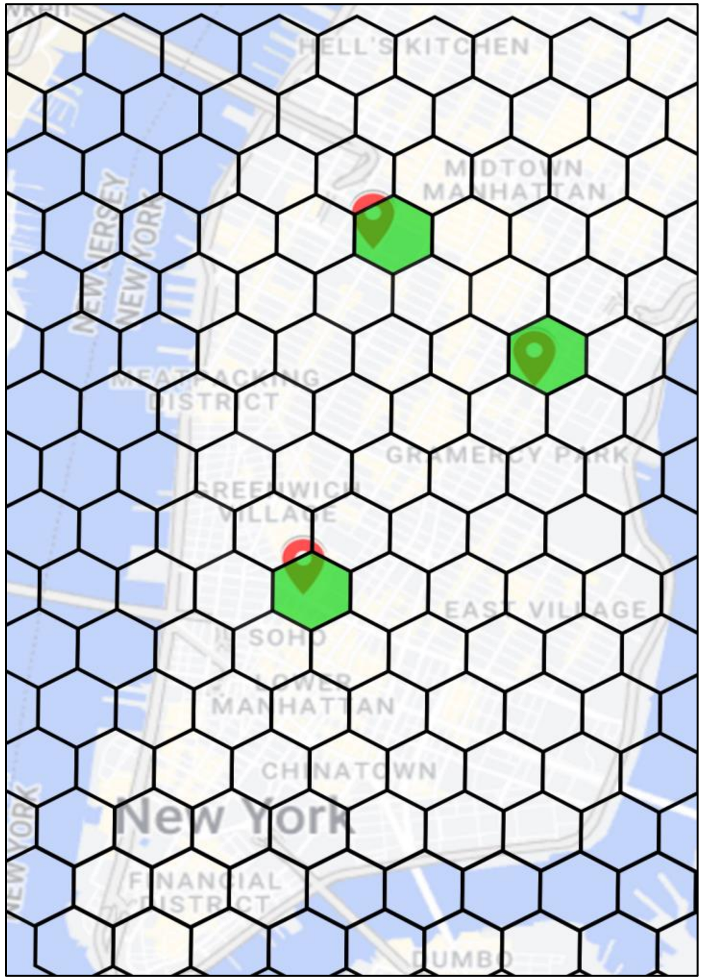


TKY

Generating higher-order check-ins



Check-ins Trajectory



Higher-order check-ins

Translate check-ins to Higher-order

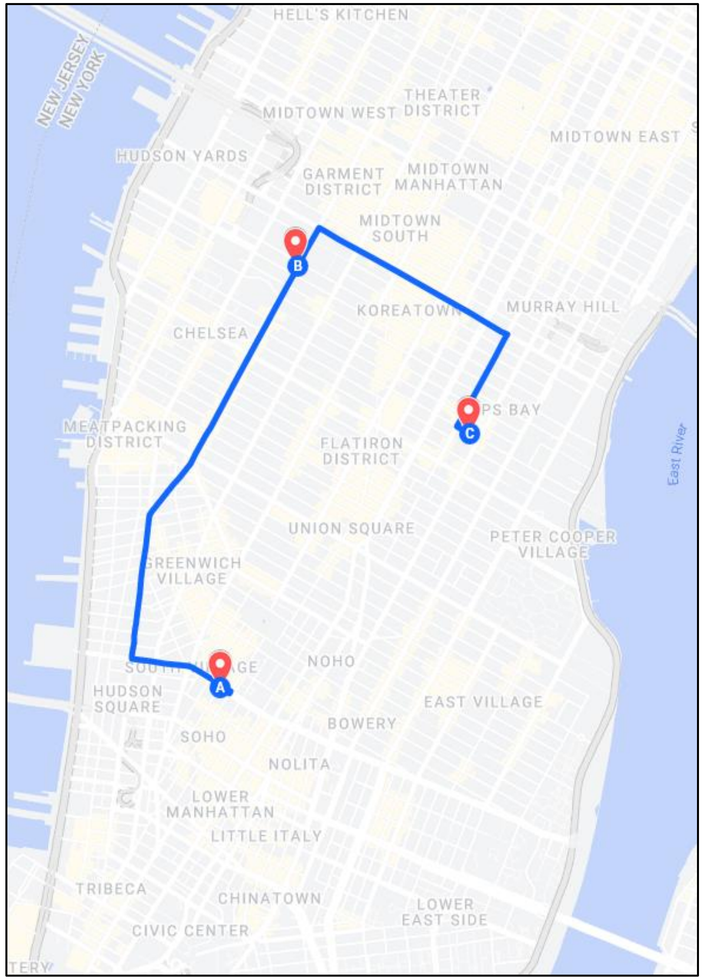
Check-ins

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

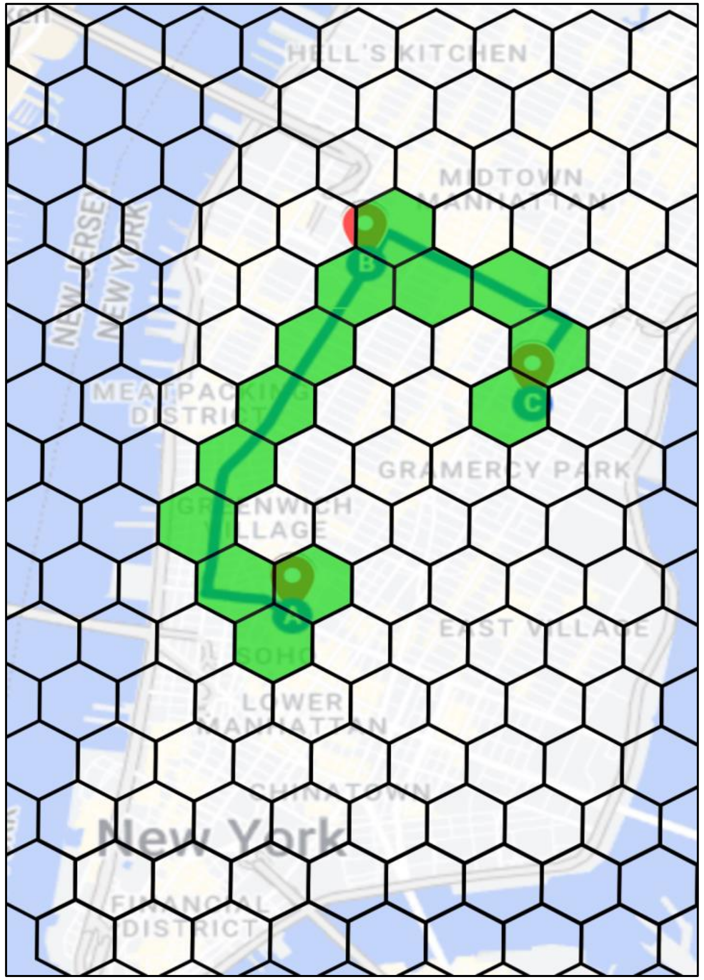
Higher-order

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

Generating Higher-order Mobility flow



Mobility flow

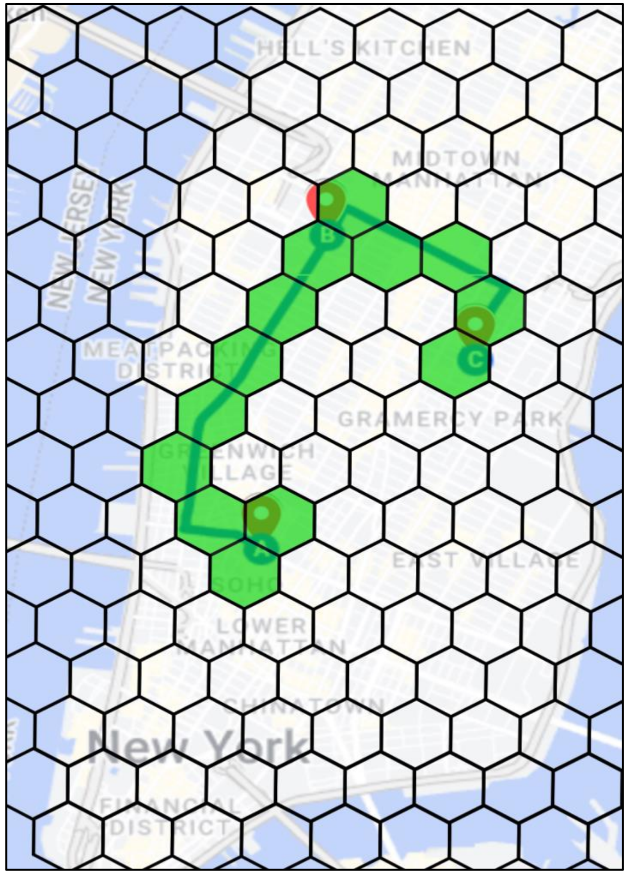


Higher-order
Mobility flow

Higher-order Mobility flow data

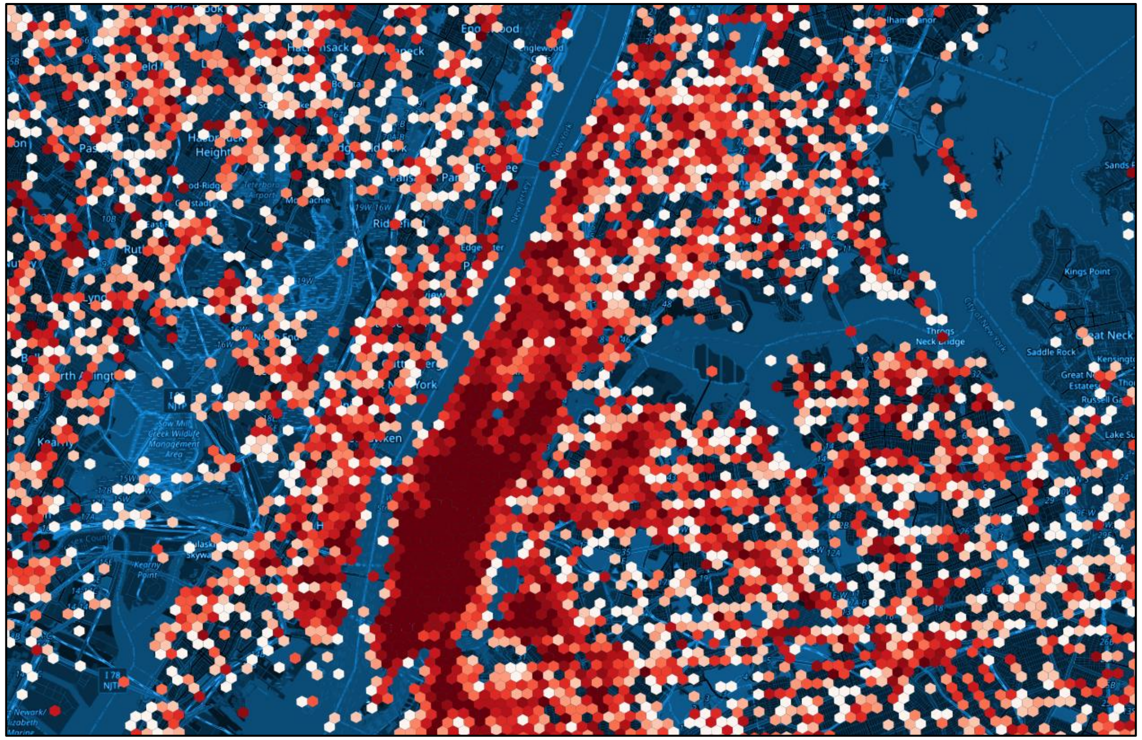
- Represented as a sequence of grid cells

- $\{g_1, g_2, \dots .\}$

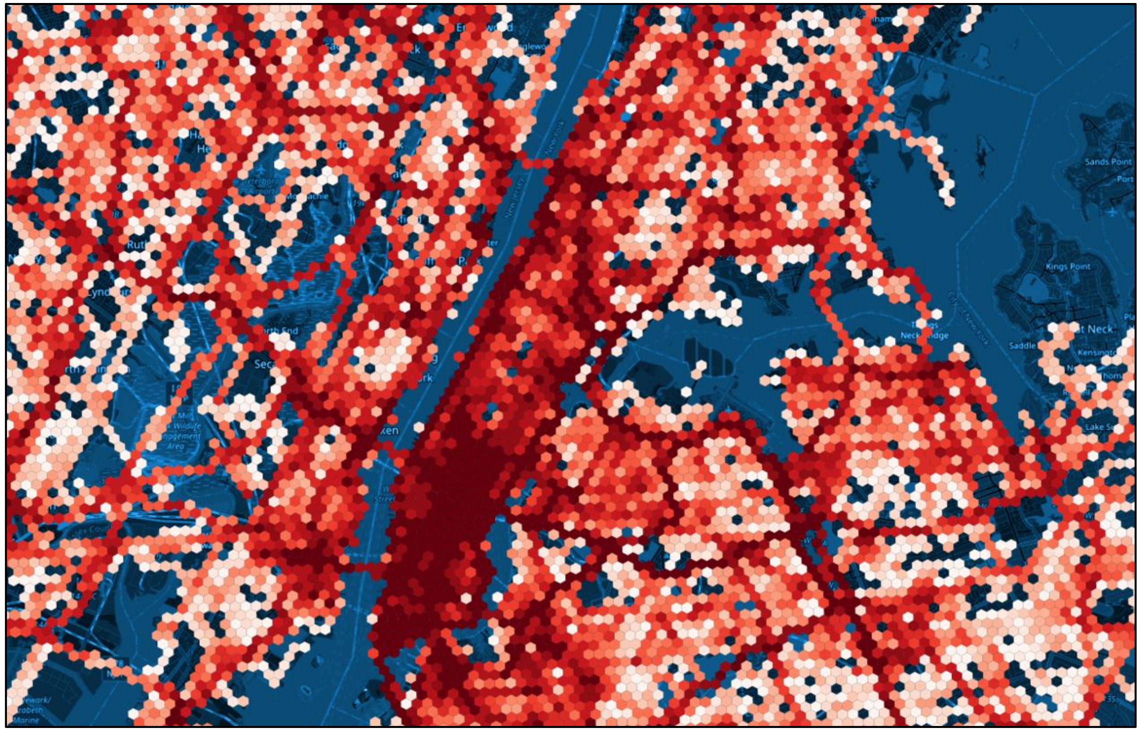


Higher-order
Mobility flow

FOURSQUARE-NYC Heatmap



Higher-order check-ins



Higher-order Mobility flow

How to calculate Sparsity ?

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

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

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}

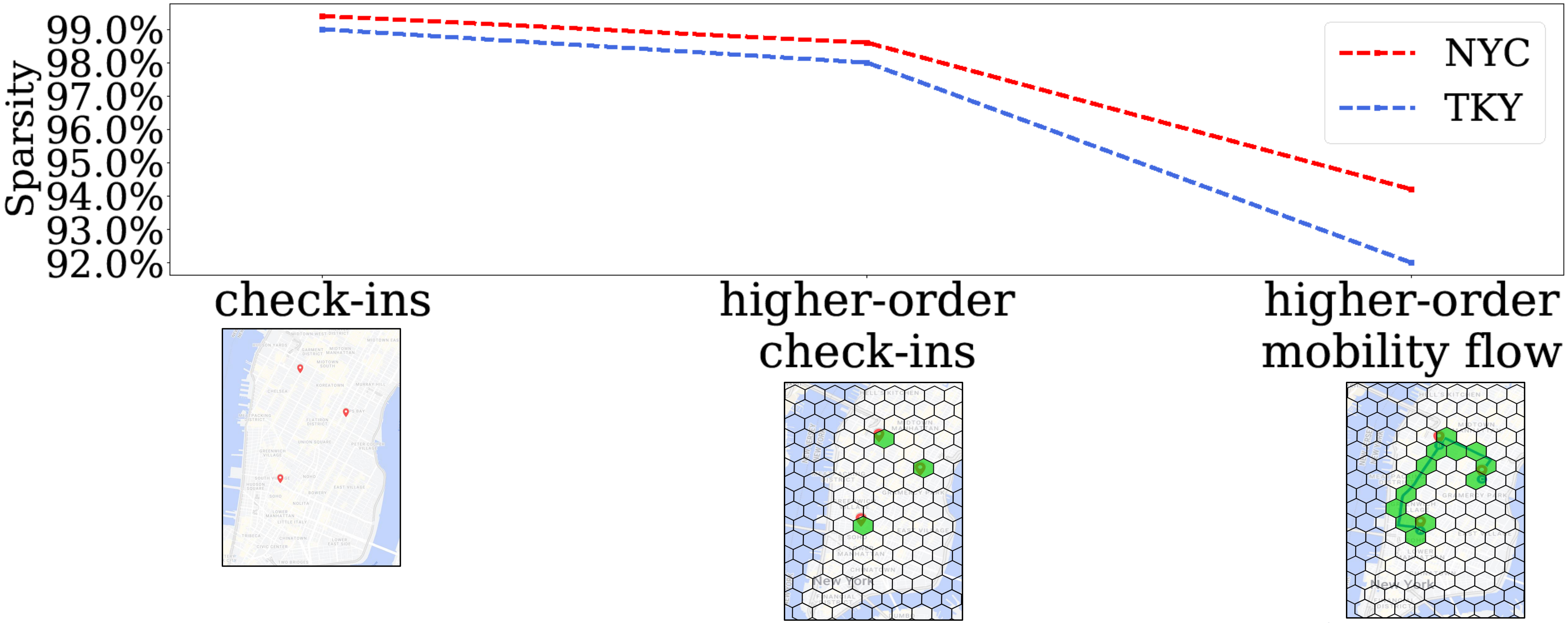
$\{g1, g2, g2\}$

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

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

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

Impact of higher-order abstraction on sparsity

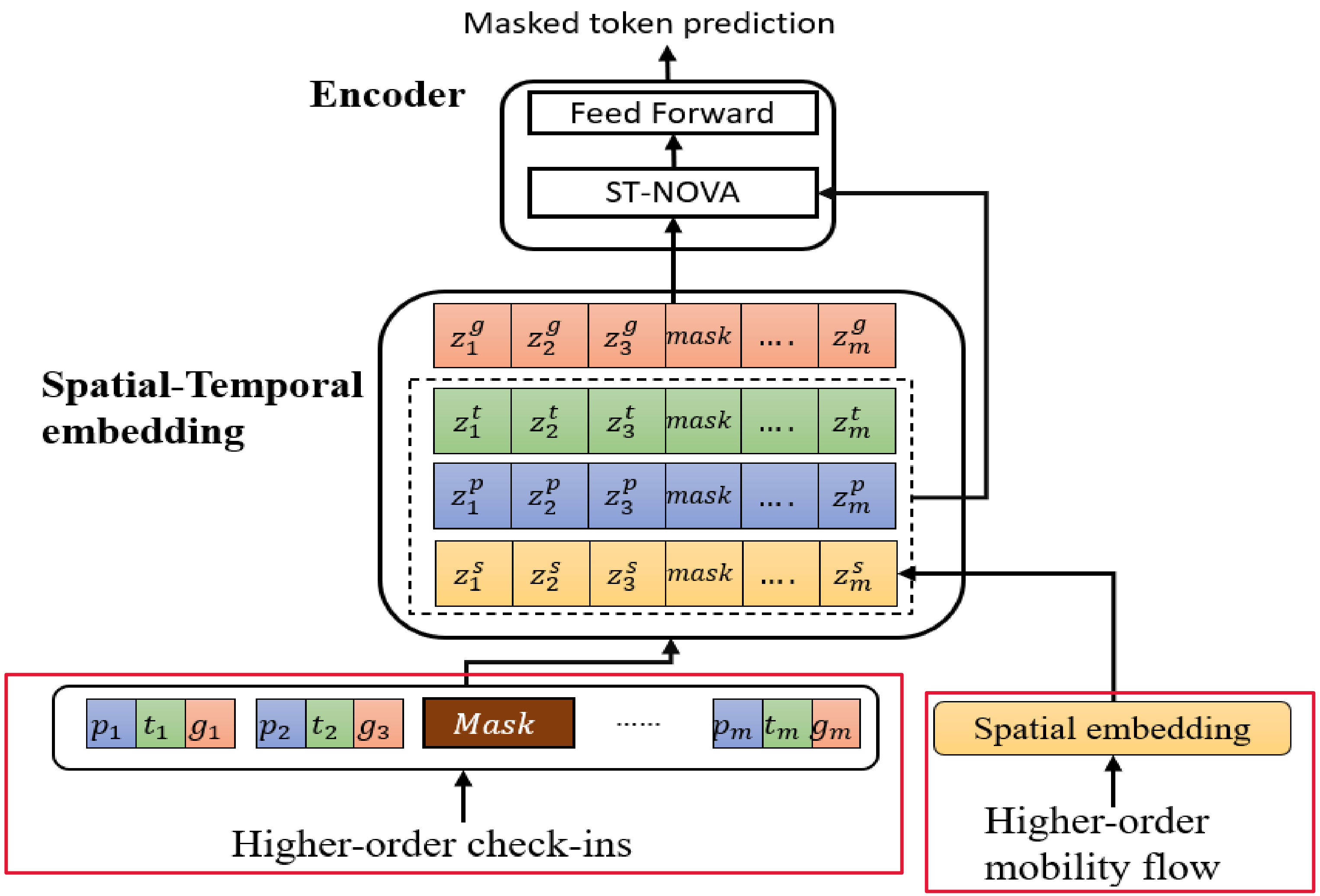


Overview

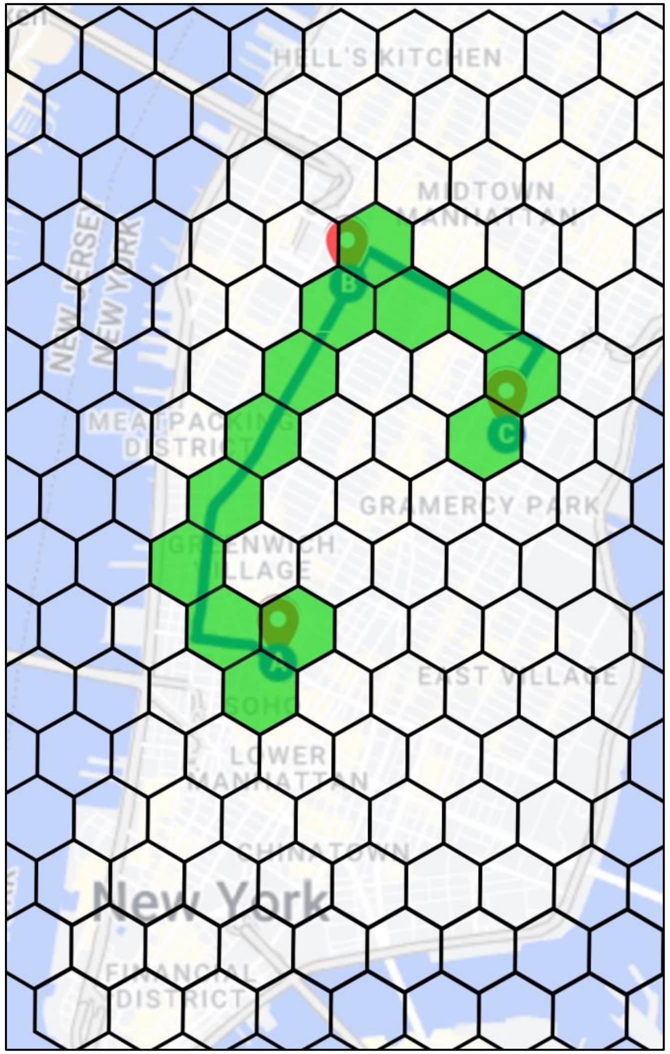
- Generating higher-order mobility flow representations
  - generating mobility flow data from check-ins
  - generating higher-order mobility flow and check-ins

- Modeling trajectory-User linking

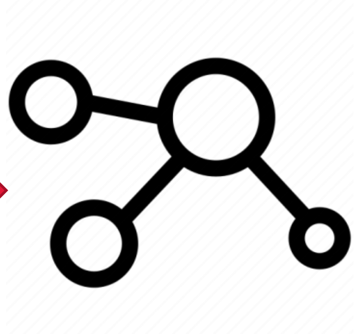
TULHOR (trajectory-user linking using higher-order representations)



Spatial embedding

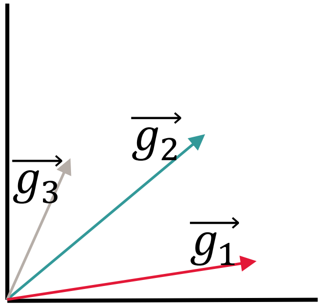


Higher-order
Mobility flow



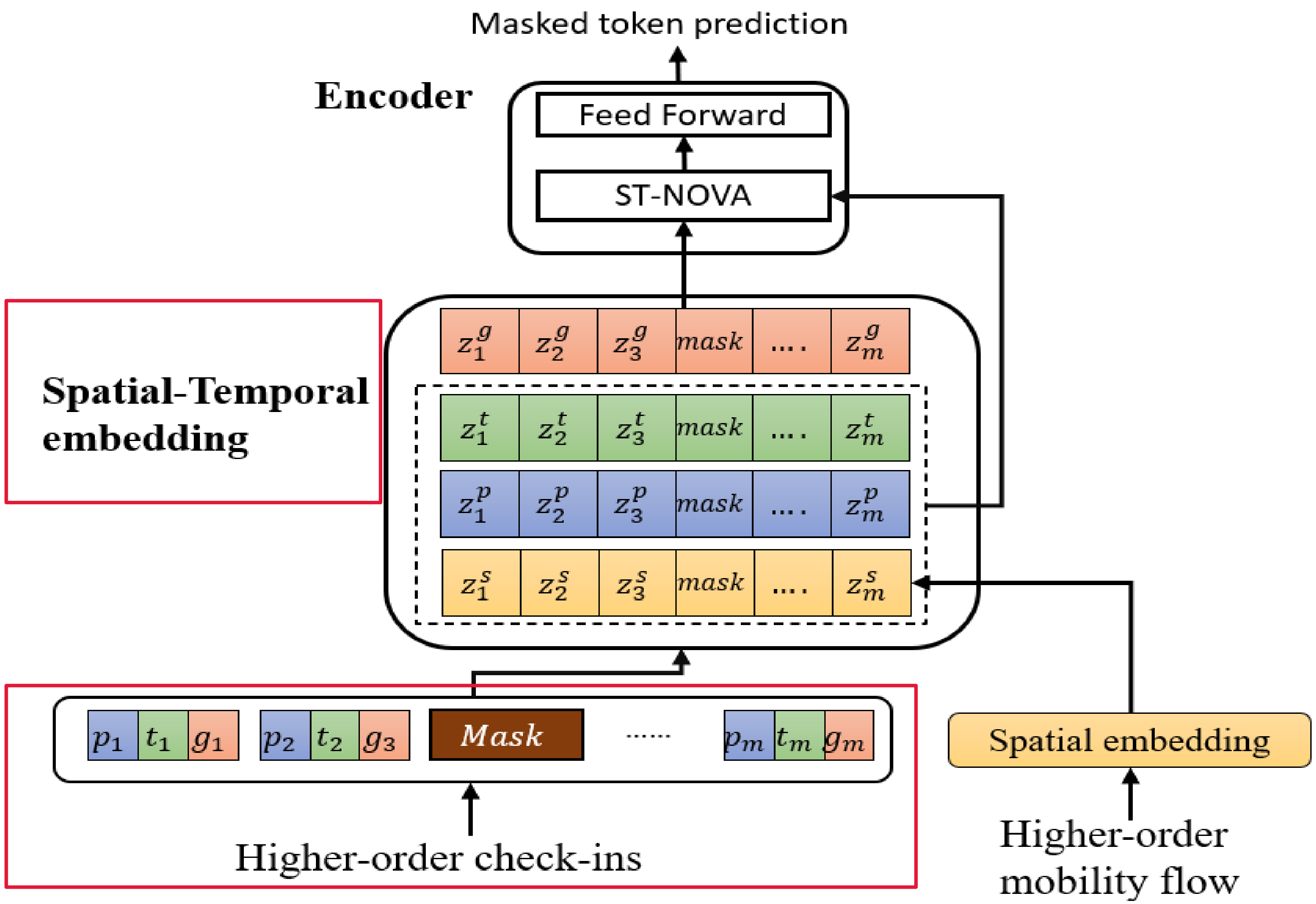
Mobility flow graph

Node2Vec



Spatial
Embedding

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



Spatial-temporal embedding

Higher-order check-ins $\{(\boxed{p_1}, t_1, \boxed{g_1}), (\boxed{p_2}, t_2, \boxed{g_2}), \ldots \ldots, (\boxed{p_m}, t_m, \boxed{g_m})\}$

$$z_i^{g} = \phi_g(g_i, W_g)$$
$$z_i^{p} = \phi_p(p_i, W_p)$$
$$z_i^{s} = \phi_s(g_i, W_s)$$

Temporal-aware positional encoding

$$\{(p_1, \boxed{t_1}, g_1), (p_2, \boxed{t_2}, g_2), \ldots \ldots, (p_m, \boxed{t_m}, g_m)\}$$

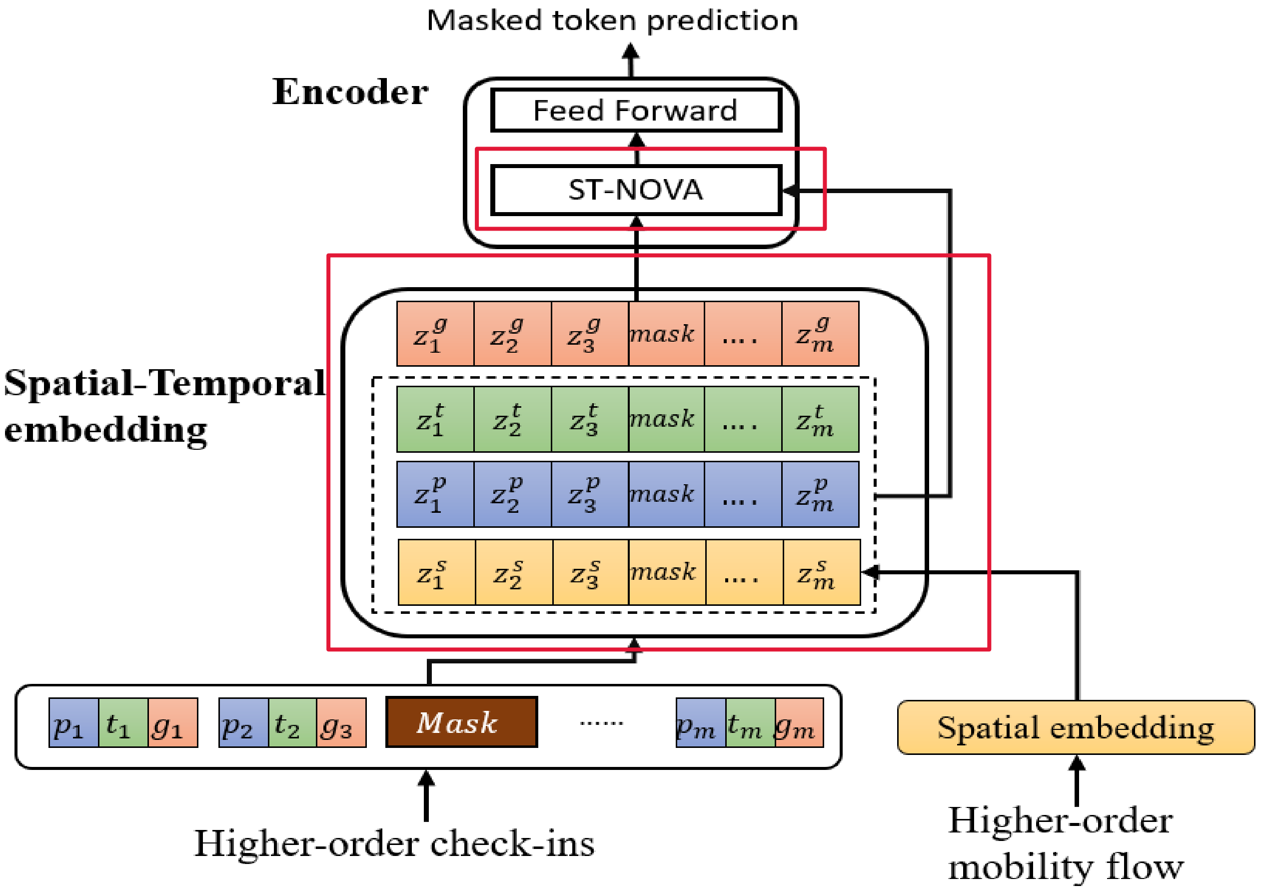
$$[z_i^t]_j = \begin{cases} \sin(w_j t_i), & \text{if } j \text{ is odd} \\ \cos(w_j t_i), & \text{if } j \text{ is even} \end{cases}$$

Output of the Spatial-temporal embedding

$$R^{(id)} = z_1^g, z_2^g, ..., z_m^g$$

$$R = (\{z_1^s, z_2^s, ..., z_m^s\}, \{z_1^p, z_2^p, ..., z_m^p\}, \{z_1^t, z_2^t, ..., z_m^t\})$$

TULHOR



Masked token prediction
Encoder
Feed Forward
ST-NOVA
Spatial-Temporal embedding
z_1^g z_2^g z_3^g mask .... z_m^g
z_1^t z_2^t z_3^t mask .... z_m^t
z_1^p z_2^p z_3^p mask .... z_m^p
z_1^s z_2^s z_3^s mask .... z_m^s
p_1 t_1 g_1 p_2 t_2 g_3 Mask ...... p_m t_m g_m
Spatial embedding
Higher-order check-ins
Higher-order mobility flow

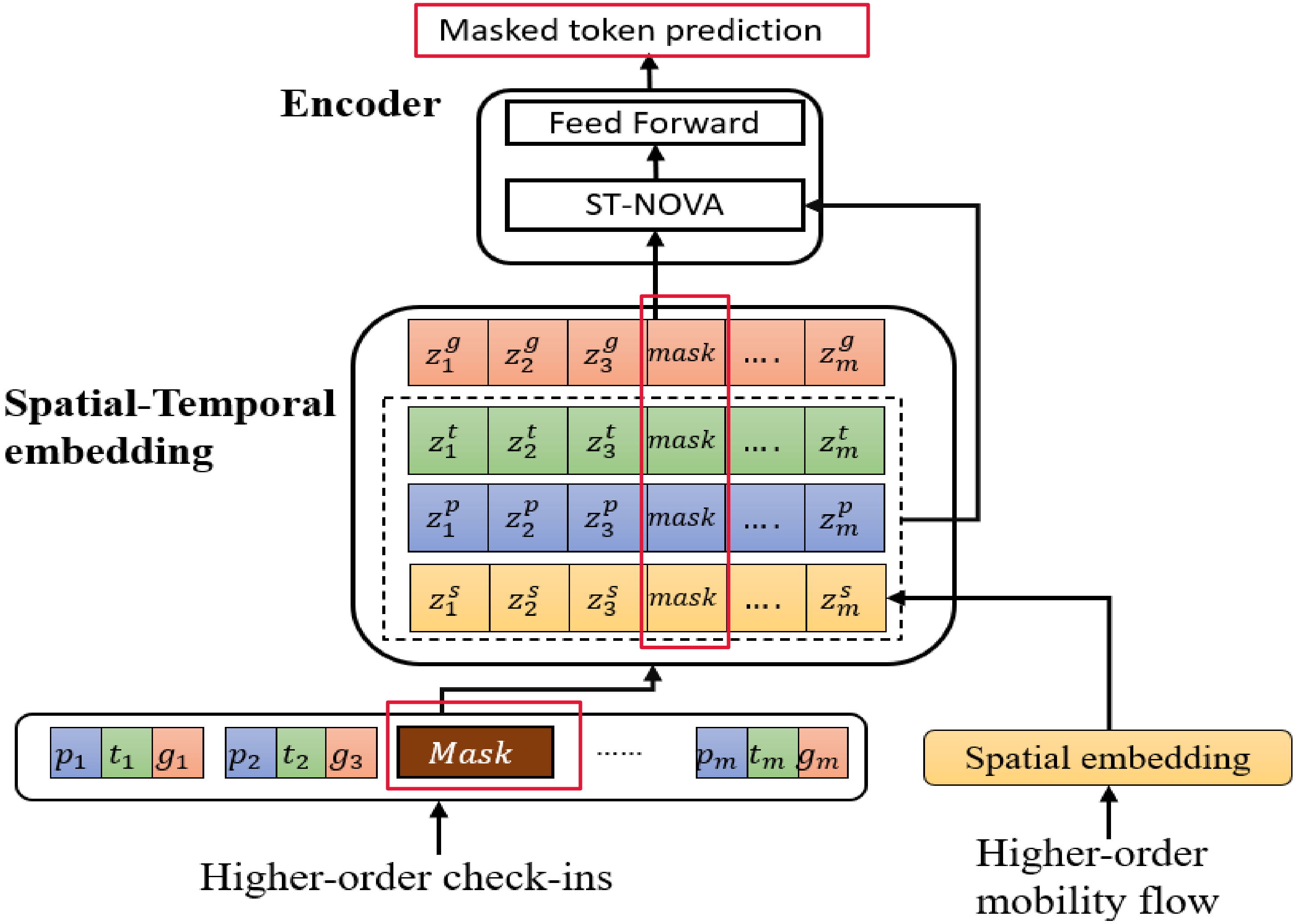
ST-NOVA

$$NOVA(R^{(id)}, R) = \sigma(\frac{QK^T}{\sqrt{d_L}})V$$

$$Q = R^{id} \times W_Q, K = F \times W_k, V = F \times W_V$$

$$F = MLP(R^{(id)} || R)$$

TULHOR



Masked token prediction
Encoder
Feed Forward
ST-NOVA
Spatial-Temporal embedding
z_1^g z_2^g z_3^g mask .... z_m^g
z_1^t z_2^t z_3^t mask .... z_m^t
z_1^p z_2^p z_3^p mask .... z_m^p
z_1^s z_2^s z_3^s mask .... z_m^s
p_1 t_1 g_1 p_2 t_2 g_3 Mask ...... p_m t_m g_m
Spatial embedding
Higher-order check-ins
Higher-order mobility flow

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

Experiment

Overview

- Datasets
  - Foursquare NYC and TKY

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

Dataset	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{T}\rvert$
Foursquare-NYC	108	6795
	209	9,637
	234	10,133
Foursquare-TKY	108	9343
	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

Model	Foursquare-TKY $\lvert\mathcal{U}\rvert = 108$					$\lvert\mathcal{U}\rvert = 209$					$\lvert\mathcal{U}\rvert = 451$				
	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%

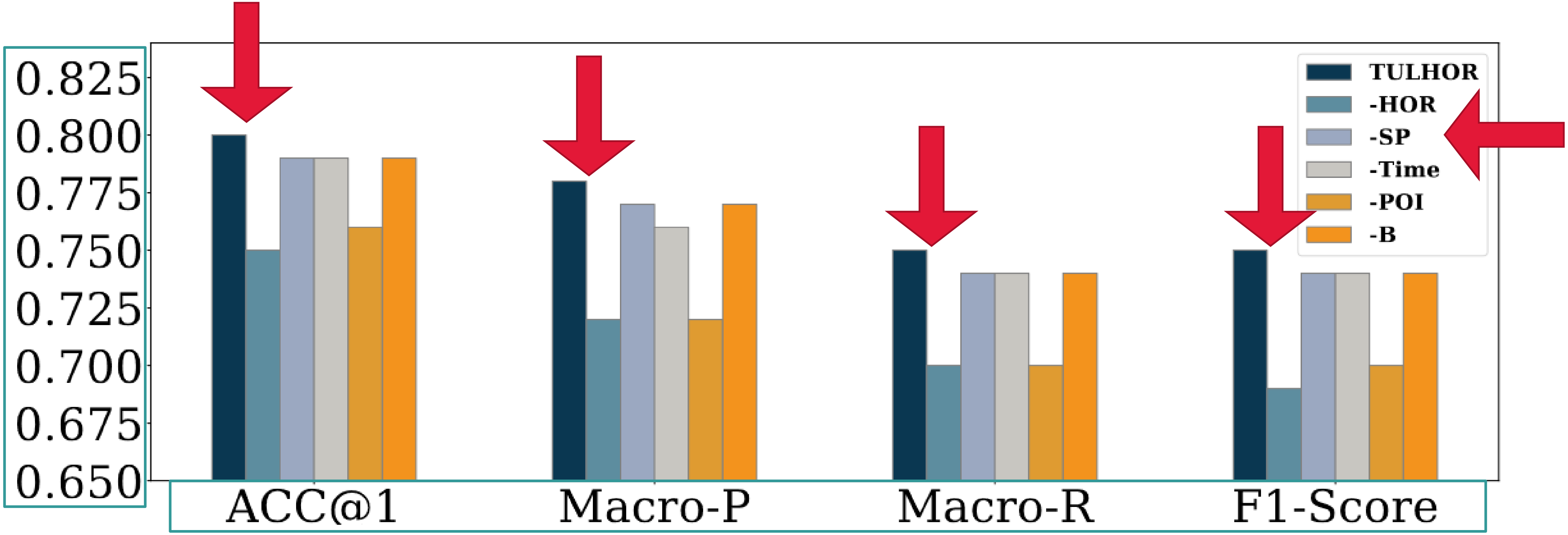
TULHOR outperforms every baseline

TULHOR has better scalability

LASSONDE SCHOOL OF ENGINEERING | YORK U

	Foursquare-NYC														
	$\|\mathcal{U}\| = 108$					$\|\mathcal{U}\| = 209$					$\|\mathcal{U}\| = 234$				
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

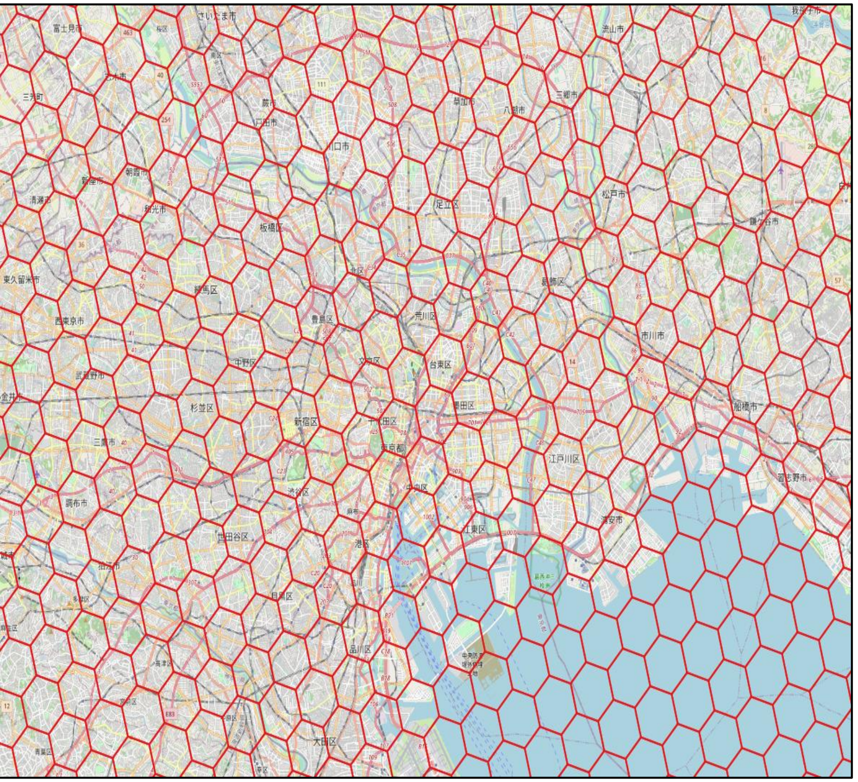


Removing Higher-order significantly reduces the performance

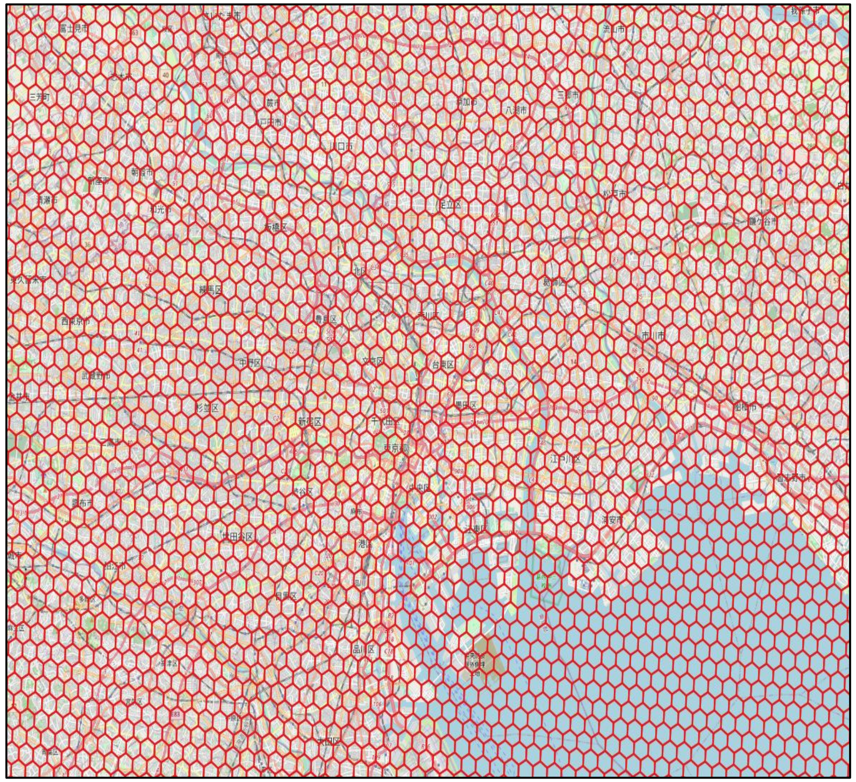
Tessellation granularity (grid size) effect

Resolution	# of cells	cell size (km^2)
Hex@7	334	5.160
Hex@8	2,003	0.730
Hex@9	11,036	0.015

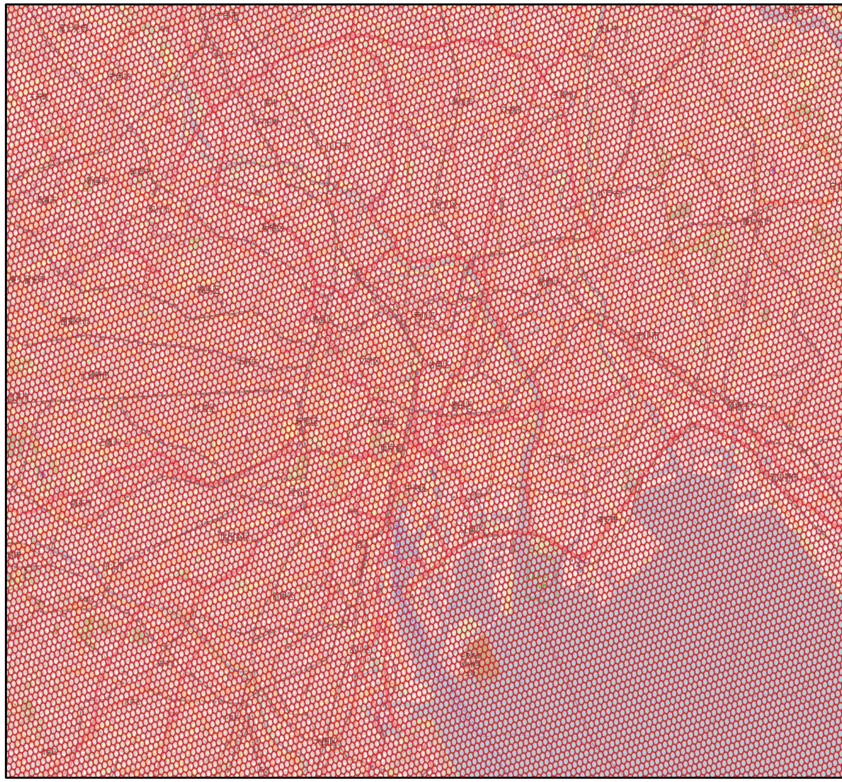
Tessellations of Tokyo



Hex@7



Hex@8



Hex@9

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	<u>0.943</u>	0.832	0.817	0.815	0.711	0.883	0.734	0.734	0.711
HEX@8	<u>0.926</u>	**0.977**	<u>0.925</u>	<u>0.917</u>	<u>0.917</u>	<u>0.868</u>	0.940	<u>0.862</u>	<u>0.849</u>	<u>0.849</u>	<u>0.790</u>	<u>0.884</u>	<u>0.753</u>	<u>0.740</u>	<u>0.733</u>
HEX@9	**0.939**	<u>0.973</u>	**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

LASSONDE SCHOOL OF ENGINEERING | YORK U

Computing Environment

- Implemented in Python

- Run in a GPU cluster (lion)

Conclusion

Summary

- Method for generating higher-order mobility flow & check-ins
- Novel deep learning framework – TULHOR (trajectory-user linking using higher-order representations)
- Extensive set of experiments

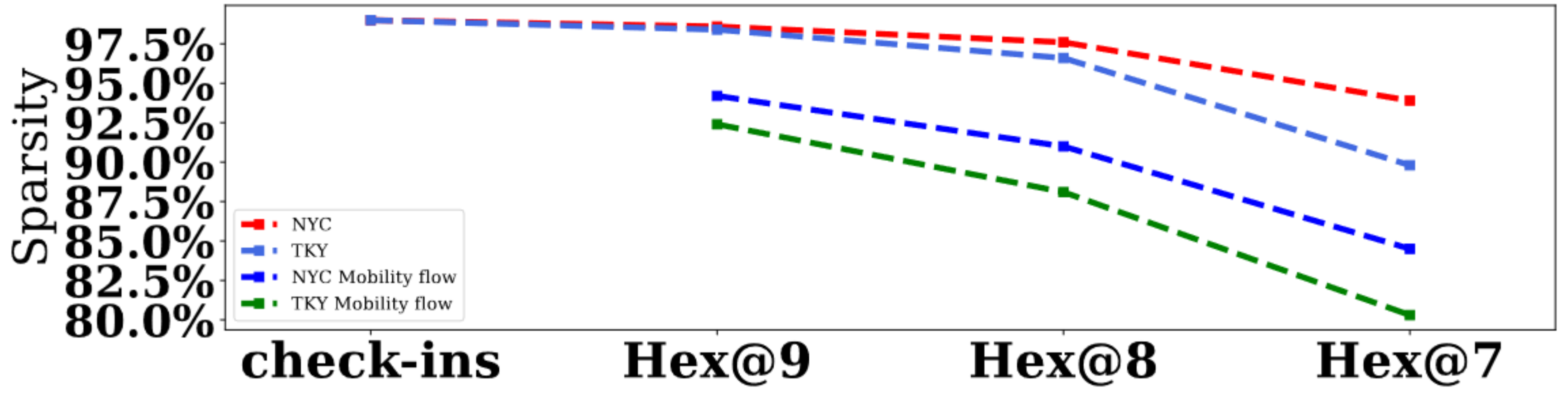
Contributions

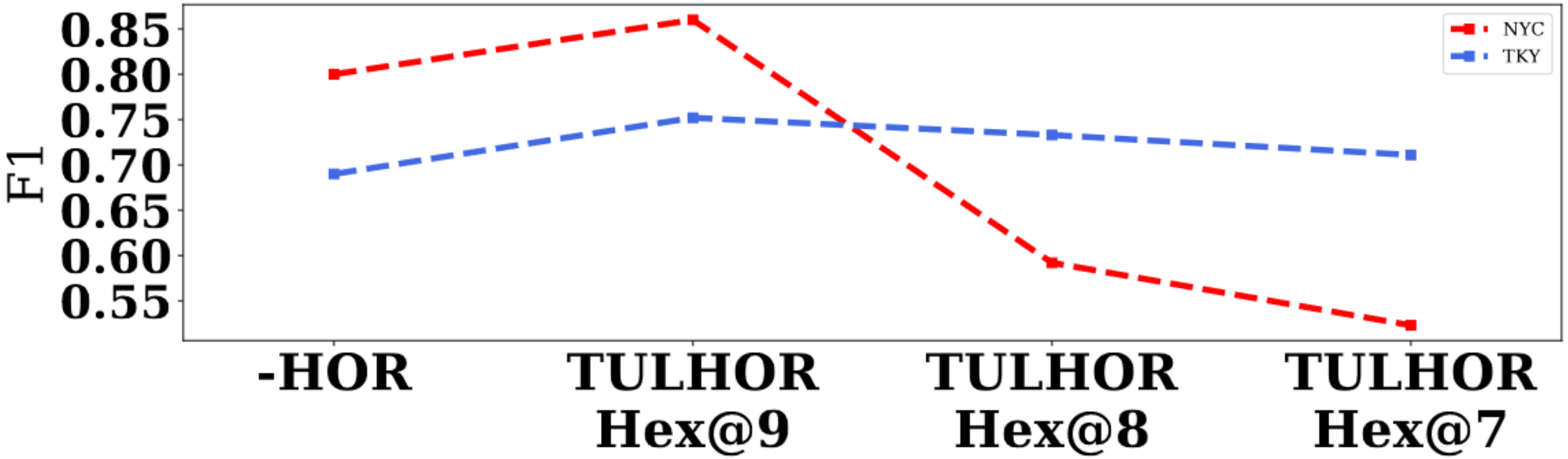
- Model agnostic method for dealing with sparsity and low data quality
- Implementation of multiple TUL models
- Publicly available dataset
- Accepted to The 24th IEEE International Conference on Mobile Data Management (MDM)

Future works

- Multi-trajectory User Linking
- POI recommendation
- Mixed hierarchical representation

Impact of Sparsity





LASSONDE SCHOOL OF ENGINEERING | YORK U

Balanced loss

$$\boxed{\mathcal{L}(Tr, u_i)} = \boxed{\frac{1-\beta}{1-\beta^{n_{u_i}}}} \boxed{log(\sigma(y'))}$$