**Real Model Steps:**

To generate real model embedding’s and visualize it using t-SNE we create a *table (traj\_as\_cells)* using postgis.

For around 1000 trajectories and 1225 cells, it takes 3 min 31 secs.

This table stores the *trajectory* and *cell* information through which traj is passing through.

1000 randomly sampled trajectories are stored in traj\_as\_cells\_ny\_sample\_1000.csv

This table is then read by the script ***real\_model\_graph.py***that outputs walks i.e. a sequence of *lists* containing *cell ids* for each trajectory through which it is passed through.

*k* perturbations are added from code: *shuffle\_walks.py*

These walks are given to another script ***real\_model\_main.py*** which is the modified version of *node2vec* that avoids generating random walks and accepts the given real walks of trajectories as random walks. This script generates the embedding.

***real\_model\_main.py*** is executed through cmd by using slightly different command then *node2vec.*

*python3 real\_model\_main.py --input walks.txt --output real\_nodes.emb*

Further these embedding are given to t-SNE (script ***tsne-vis.py***) for visualization.

**Null Model Steps:**

To generate null model embedding’s and visualize it using t-SNE we will generate a graph of grid cells. Each cell is a node and its connected to its adjacent cells by an edge.

The above graph is generated by script ***embeddings\_tsne.py*** which outputs an *edgelist* of cell ids in a format accepted by *node2vec*.

***Node2vec*** generates embedding’s which are given to t-SNE (script ***tsne-vis.py***) for visualization.

**Cosine Similarity:**

To calculate cosine similarity, embedding files (both real and null model) are given to script ***getting\_vectors\_cosine\_sim.py*** which stores cosine similarities and their difference for both models in a file.

File (*cos\_sim\_10\_walks.csv*) contains comparison between real and intermediate model.

Made separate files for real, null and diff as: real\_cos\_10, null\_cos\_10

And (*cos\_sim.csv)* contains comparison between real and null model.

Made separate files for real, null and diff as: real\_cos, null\_cos

**Diff (Matlab folder)**

*diff\_10.csv* contains cosine similarity difference between real and inter **Avg:** 0.157

*diff.csv* contains cosine similarity difference between real and null **Avg:** 0.139

**Cosine similarities for 10000 (ten thousand trajectories):**

* File (*cos\_sim\_ten.csv*) contains comparison between null and real model
* *cos\_sim\_ten\_10.csv* contains comparison between intermediate and real model

**Diff (Matlab folder)**

*diff\_ten\_10.csv* contains cosine similarity difference between real and inter **Avg:** 0.1625

*diff\_ten.csv* contains cosine similarity difference between real and null **Avg:** 0.133

**Getting towards clean cosine similarities.**

We generate a file (*clean\_plots*) that contains real, inter and null cosine similarities cleaned and based on the nodes that are part of all the models.

**Code:** top\_k\_cos\_diff.py

***function*** *sample\_nodes()* get the nodes that are part of real model.

***function*** *sample\_cos()* we take the sample nodes of real model and get the same nodes from  
intermediate and null model and store it in the file plot\_nodes.csv.

***function*** *clean\_plots()* this function removes the rows that contains ‘less’ and ‘infinite’.

Further we reverse sort the cosine similarities generated by the 3 models and store them in separate files to generate plots using MATLAB.

**Walks (New York):**

***Real Model:*** This model is generated by taking walks of real trajectories and then generating k perturbations of those trajectories, where ­­.

**1000 traj** **walks:** *walks.txt* and *walks\_10.txt*  with k perturbations  
Generating k perturbations of the same trajectory makes it more comparable to the null model. As we are taking 10 random walks per node in null model.

*real\_nodes.emb* contains embeddings for 1000 trajectories

*real\_nodes\_10.emb* contains embeddings for 1000 trajectories with k perturbations

**10,000 traj walks:** *walks\_ten.txt and walks\_ten\_10.txt* with *k* perturbations

*Code file: shuffle\_walks.py (****location:*** *embeddings\walks\)*

­­*real\_nodes\_ten\_10.emb* contains embeddings for 10,000 traj with k perturbations

***Intermediate Model:*** This model is closer to real model as compared to null model. We generate the following walks.

**1000 traj walks:**  *walks (walks\_inter.txt – with k perturbations*) are generated by keeping first node as similar to real and the rest of nodes are taken at uniformly randomly from the neighboring nodes. We keep the size of per walk as (α) by taking the average from the real walks.

*walks\_inter.emb* contains the embeddings

*Code file: shuffle\_walks.py (****location:*** *embeddings\walks\)*

**10,000 traj walks:**

*walks\_inter\_10.txt* contains walks with k perturbations.

*walks\_inter\_10.emb* contains embeddings

***Null Model:*** This model is based on the random walks generated by node2vec. We simply give the network to node2vec and it generates walks and embeddings of grid nodes. To make it comparable to real model, we take average of real walks and based on that we keep the size of . (it varies based on avg of real walks)

We also take the walks and generate *k* perturbations to make it comparable to intermediate and real model walks. (random\_walks\_10.txt)

*Code file: shuffle\_walks.py (****location:*** *embeddings\walks\)*

**Trajectories:**

By fetching the first 10 taxi points from train.csv and generating trajectories I made sure that we are experimenting with the trajectories created from first 1000 taxi pickup and drop off points.

**Trajectory Sampling:**

We uniformly randomly sample 1000 trajectories pick\_up and drop\_off points from *train.csv* and then generate trajectories out of them. These are the stored in *random\_1000\_traj.csv.*

Further we sampled 10000 trajectories from train.csv and generated *traj\_as\_cells.csv* out of them.

We generated walks (real\_model\_graph.py) from those and are storing them in file *walks\_fast.txt.*

*walks\_fast.emb* contains embedding for 10,000 trajectories.

**Deletion:**

Trajectories with one coordinate gets removed.

**Trajectory Appearance**

Many trajectories walks don’t appear in the real walks, as they are out of the grid area.

**Creating Plots**

Number of real cosine similarity nodes are the same in both *(real\_intermediate)* and *(real\_null)* comparison.

**Matlab folder**

*clean\_plots.csv* (contains cosine similarity for all models for 1000 trajectories)

*clean\_plots\_ten.csv* (contains cosine similarity for all models for 10,000 trajectories)

*diff.csv* (real and null) *and diff\_10.csv* (real and inter)contains cosine similarity differences for (1000 trajectories)

*diff\_ten.csv* (real and null) *and diff\_ten\_10.csv* (real and inter)contains cosine similarity differences for (10,000 trajectories)

**City of Porto**

**Real**

*traj\_as\_cells\_porto.csv* contains traj and cell ids fetched from posgres. (1000 traj)

*walks\_porto.txt* contains walks for 1000 randomly sampled trajectories. *walks\_porto\_10.txt* contains walks with k perturbations.

*real\_porto.emb* contains embeddings for 1000 real trajectories.

**Null**

*random\_walks.txt* contains walks for grid 35x35. *random\_walks\_10.txt* containswalks with *k* perturbations.

*null\_porto.emb contains embeddings for 35x35 grid*

**Intermediate**

*walks\_porto\_inter.txt* contains walks for intermediate model with k perturbations

*inter\_porto.emb contains embeddings for inter model*

**Cosine Similarity**

*cos\_sim\_porto\_10.csv* contains comparison between intermediate and real model *cos\_sim\_porto.csv* contains comparison between null and real model

*clean\_plots\_porto.csv* contains cosine similarity between all three models.

**Diff**

*diff\_porto\_10.txt* contains cosine similarity difference between real and inter **Avg:** 0.166

*diff\_porto.txt* contains cosine similarity difference (real and null) **Avg:** 0.127

**10,000 trajectory**

**Real**

*traj\_as\_cells\_porto\_ten.csv* contains contains traj and cell ids fetched from posgres.

*walks\_porto\_ten* contains walk for 10,000 trajectory and *walks\_porto\_ten\_10.txt* contain walks with *k* perturbations  
*real\_porto\_10.emb* contains embedding for real model

**Intermediate**

*walks\_porto\_inter\_ten.txt* contains walks for intermediate model with k perturbations  
*inter\_porto\_ten.emb* contains embeddings for inter model

**Cosine Similarity**

*cos\_sim\_porto\_ten.csv* contains comparison between null and real model  
*cos\_sim\_porto\_ten\_10.csv* contains comparison between inter and real model

*clean\_plots\_porto\_ten.csv* contains cosine similarity between all three models.

**Diff**

*diff\_porto\_ten.txt* contains difference between real and null model **Avg:** 0.134  
*diff\_porto\_ten\_10.txt* contains difference between real and inter model **Avg:** 0.181

**HeatMap**

**Trajectory as POI (New York Data-set)**

**Null Model**

we use the above query to store all the cell and poi ids based on if they are found inside the boundary of cell. And order it by cell ids.

*cell\_poi\_ny.csv* is given to *traj\_as\_poi.py* to find cells in random walks (file *random\_walks.txt*) on grids and replace them with the poi ids.

null\_walks\_poi.txtcontain walks of pois and *null\_poi\_nodes.emb* contain embeddings

**Real Model**

We take two tables *traj\_as\_cells –* in which trajectories are represented as *cells* and *cell\_poi\_ny –* in which cells are represented as *pois* and get *traj\_as\_pois*

*traj\_as\_poi\_ny\_1000.csv* contains trajs and pois associated with them.

*walks\_poi.txt* contains walks for real model and *poi\_nodes.emb* contains embeddings.