**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.

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

**Walks:**

***Intermediate Model:*** This model is closer to real model as compared to null model. We generate the following walks.*Shuffled walks (shuffled\_walks.txt*) 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 20 by taking the average from the real walks.

***Real Model:*** This model is generated by taking walks of real trajectories and then generating k perturbations of those trajectories, where ­­. File name is *(shuffled\_walks\_10.txt).*  
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.

***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.

**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.*

**Deletion:**

We had to delete couple of trajectories as they had only one coordinate.

* 994 trajectories

**Trajectory Appearance**

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