I Open at the Close:  
A Deep Reinforcement Learning Evaluation of Open Streets Initiatives

R. Teal Witter*, Lucas Rosenblatt*

New York University
{rtealwitter, lucas.rosenblatt}@nyu.edu

Abstract

The open streets initiative “opens” streets to pedestrians and bicyclists by closing them to cars and trucks. The initiative, adopted by many cities across North America, increases community space in urban environments. But could open streets also make cities safer and less congested? We study this question by framing the choice of which streets to open as a reinforcement learning problem. In order to simulate the impact of opening streets, we first compare models for predicting vehicle collisions given network and temporal data. We find that a recurrent graph neural network, leveraging the graph structure and the short-term temporal dependence of the data, gives the best predictive performance. Then, with the ability to simulate collisions and traffic, we frame a reinforcement learning problem to find which streets to open. We compare the streets in the NYC Open Streets program to those proposed by a Q-learning algorithm. We find that the streets proposed by the Q-learning algorithm have reliably better outcomes, while streets in the program have similar outcomes to randomly selected streets. We present our work as a step toward principally choosing which streets to open for safer and less congested cities.

Introduction

Traffic congestion is at its best inconvenient and at its worst very dangerous. In 2022, American drivers spent an average of 51 hours stuck in traffic, representing an estimated 81 billion dollars in productivity loss (INRIX 2022). Furthermore, a by-product of increased traffic in urban areas is an increase in traffic collisions (Retallack and Ostendorf 2019). Both traffic and collisions can be mitigated by intelligent road network design, but urban road networks are already built and new infrastructure projects in cities can be prohibitively expensive (Siemiatycki 2015).

One potential solution is to “open” existing roads to pedestrians and bicyclists by closing them to cars and trucks (Kuhlberg et al. 2014; Bertolini 2020). Generally, open streets initiatives provide a communal space for people living in urban environments. As expected, the initiatives have positive impacts, including on the physical health of the participants (Cohen et al. 2016; Sharples 2014). But there is some evidence that opening streets also improves traffic and safety. Two prominent examples are Times Square and Herald Square in Manhattan, NYC, which were turned into pedestrian plazas in 2009. After the squares opened, there was a reduction of approximately 15% travel times for routes along Broadway and a 63% reduction in injuries to drivers and passengers on the avenue (Grynbaum 2010).

Numerous exogenous variables make an empirical analysis of the effects of opening streets challenging. As a result, cities generally use a proposal process to identify which streets to open. Identifying candidate streets for opening has been studied (Youn, Jeong, and Gastner 2008; Rhoads et al. 2021); however, previous work does not simulate the effects of proposed open streets nor does it systematically consider the impact of open streets on public safety and vehicular congestion. In this work, we do both.

Contributions

In the first part of our work, we build an improved model for predicting collisions, evaluated on granular and comprehensive data. We consider a wider time frame (days, months and years) and a larger space (the entirety of Manhattan) for predicting collisions than prior work. We are the first to: (1) use years of data that account for seasonal and annual variations in traffic and exogenous variables like weather (prior work only used several months), (2) use all negative and positive examples (prior work subsampled to enforce class balance), and (3) take a global view of the road network, predicting collisions at the city level while taking into account local information (prior work used only a few block radius around a collision). We compare several models for the prediction task. Our best model uses recurrent layers to capture short-term temporal dependencies and graph convolutional layers to capture spatial dependencies of our data. Finally, we analyze the importance of features in the best model and discuss connections to prior work in the transportation literature.

In the second part of our work, we use a deep learning approach to evaluating the efficacy of the NYC Open Streets program. To the best of our knowledge, we are the first to formulate the problem of opening roads in the language of reinforcement learning. We simulate road openings in real historical days. For each simulated day, we estimate traffic as the total car density per capacity of each
road and estimate collision risk using our collision prediction model from the first part of our work. We train a deep Q-learning model to output the long term value of opening each road segment. We find that the streets opened by the NYC Open Streets program have similar performance to randomly selected streets and that the streets in the program are geographically concentrated in certain Manhattan neighborhoods. In contrast, the streets with the highest Q-values consistently reduce collisions and traffic, with a more equal distribution across Manhattan. As a result, we recommend using our Q-learning model as an additional method for evaluating streets in the NYC Open Streets program.

Figure 1 summarizes the two parts of our work. All of our code and data are available on Github.²

Related Work

Open Streets Open street initiatives are multifaceted in function: they allow more space for people to safely exercise and traverse the city, they promote a decrease in vehicle traffic and carbon emissions, and they provide expanded outdoor space for businesses, particularly restaurants (Hazarka 2021). However, the initiatives have faced some challenges. Critics have pointed out a lack of equity in the implementation, arguing that more affluent neighborhoods have benefited disproportionately from the program in part because streets are chosen through a proposal process (Hazarka 2021). Prior work has used a coarse approximation of a city to suggest which roads to open (Youn, Jeong, and Gasteiner 2008) and suggested roads based on sidewalk infrastructure (Rhoads et al. 2021). But we are not aware of any work that simulates the effects of opening streets or systematically considers the impact on safety.

Collision Prediction Many papers have used traditional machine learning techniques to solve the problem of predicting collisions (Cheng and Koudas 2019; Baloch et al. 2020; Auret and Aldrich 2012). However, there are complicated spatial and temporal dependencies in road networks and collision dynamics. To fit these non-linear patterns, there has been substantial interest in deep learning techniques to predict collisions (Lin et al. 2021). In Ren et al. (2017), they use the recurrent long-short-term memory architecture to capture temporal relationships. In Zhao et al. (2019), they use standard convolutional layers to capture both temporal and spatial relationships. Our work is most similar to Yu et al. (2021). They use taxi, weather, road infrastructure, and point data to predict collisions with a deep graph neural network which captures both temporal and spatial relationships. However, they only consider a two-month period and downsample the number of events to equalize the number of collisions and non-collisions, reducing data-scale drastically.

Traffic Prediction The success of our work is predicated on inferring traffic flow from available taxi trip data as accurately as possible. Yu et al. (2021) focused on the prediction of collisions in Beijing, where data on the exact location of the entire Beijing taxi fleet is available in increments of 5 minutes. However, in our setting in Manhattan, we do not know the exact location of taxis during their trips. That said, we do not have access to trip start and end GPS coordinates for the period from 2013 to 2015, and we rely on this data for traffic inference (Taxi and Commission 2022).³ We note that the NYC taxi trip data before 2016 have been successfully used in a variety of applications such as fleet dispatching and routing (Bertsimas, Jaillet, and Martin 2019; Deri,

---

²https://github.com/rtealwitter/OpenStreets

³Coordinates have not been reported since 2015 due to privacy concerns.
Background

Below we summarize the notation and ideas underlying our use of graph neural networks and reinforcement learning.

Graph Neural Networks Graph Neural Networks (GNNs) are a popular choice for exploiting the structure in graphs (Welling and Kipf 2016), and a set of directed edges $\mathcal{E}$. In our work, the nodes $\mathcal{V}$ represent the segments of the road and the edges $\mathcal{E}$ represent the intersections that connect them.

Local structure is captured in a GNN through graph convolutional layers. Consider a node $v \in \mathcal{V}$ and its representation $x^f_v \in \mathbb{R}^d$ at the $f$th layer of a GNN. Then we build its representation $x^{f+1}_v \in \mathbb{R}^d$ at the next layer by applying a graph convolution with parameters $\Theta \in \mathbb{R}^{d \times d}$. In particular,

$$x^{f+1}_v = \sigma \left( \sum_{u \in \mathcal{E}} w_{u,v} x^f_u \right)$$

where $w_{u,v}$ is the normalized weight of the edge between $u$ and $v$ and $\sigma$ is an activation function.

An advantage of GNNs in our setting is that GNNs learn weights that can be applied to exploit connections in any graph, provided the node features stay consistent. The size of the graph changes in our reinforcement learning problem since each action removes a road segment. Furthermore, a GNN can capture short term temporal information by passing in a hidden state to each layer, which is important in our setting as short term temporal information is more important than long term. This architecture is called a ‘recurrent GNN’ (RGGNN) (Seo et al. 2018).

Figure 2 describes how the RGNN captures the spatial and temporal relationships in the collision prediction problem.

Reinforcement Learning Reinforcement Learning (RL) is a collection of techniques for optimization in online learning settings (Sutton and Barto 2018; Moerland et al. 2023).

The state space $\mathcal{S}$ is the set of all possible realizations of an environment while the action space $\mathcal{A}$ is the set of all possible options we can take in an environment. An agent is presented with a state and must then choose an action to take. Once an action is chosen, the agent transitions to a new state using a stochastic transition function $f : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$. The agent also receives a real-valued reward for taking an action in a given state according to a stochastic reward function $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$. A “good” policy is then one that maximizes the reward an agent receives in expectation.

In this work, we consider Q-learning because it is generally more sample efficient than other techniques and produces useful intermediate values (Jin et al. 2018). The Q function, the namesake of Q-learning, is used to find and exploit states that produce a high reward. Consider a stochastic sequence of states and actions $(s_0, a_0, s_1, a_1, \ldots)$ where $s_{i+1} = f(s_i, a_i)$ and actions are selected according to our policy. Then we can write

$$Q(s, a) = \mathbb{E} \left[ \sum_{i=0}^{\infty} r(s_i, a_i) \gamma^i \right]$$

where $0 < \gamma < 1$ is some discount factor chosen so that we focus on near term reward. If we had these $Q$ values, then our policy should choose the next action in state $s$ by calculating $\arg \max_a Q(s, a)$.

This observation motivates the Bellman equation, a natural criteria for our $Q$ function, where $s' = f(s, a)$:

$$Q(s, a) = r(s, a) + \gamma \max_{a'} Q(s', a')$$

In our setting, we will use a neural network for the $Q$ function with parameters $\theta$. Then the loss function is given by

$$\mathcal{L}(\theta) = \left( r(s, a) + \gamma \max_{a'} Q(s', a') - Q(s, a) \right)^2$$

Notice that the $Q$ function appears in two places. During optimization, we freeze weights of the `target' $Q$ function on the left and update the $Q$ function on the right. Q-learning allows us to efficiently train the $Q$ function; notice that we only need the 4-tuple $(s, a, s', r)$ to compute the loss.

Part 1: Predicting Collisions

In this section, we describe our work on predicting collisions. Compared to previous work, our work considers a
Imbalanced Classification

Collisions over a road network framing over the regression framing. In order to predict collisions, we leverage infrastructure data like road attributes, day-specific weather conditions, and traffic data. The data we use are granular: we have information for all road segments in Manhattan every day over a three-year period. Each road segment is defined as the portion of a street between two intersections, yielding 19,391 segments in Manhattan. Unfortunately, traffic data is not available at our geographic and temporal scale. Instead, we infer overall traffic in our road network using a massive set of start and end locations from taxi trips. We use Dijkstra’s shortest-path algorithm to efficiently calculate where taxis, and we assume other vehicles, likely drove. Since our data set is massive (10 to 15 million taxi trips per month), we used simplified shortest paths and local rerouting.

Data

In order to predict collisions, we leverage infrastructure data like road attributes, day-specific weather conditions, and traffic data. The data we use are granular: we have information for all road segments in Manhattan every day over a three-year period. Each road segment is defined as the portion of a street between two intersections, yielding 19,391 segments in Manhattan. Unfortunately, traffic data is not available at our geographic and temporal scale. Instead, we infer overall traffic in our road network using a massive set of start and end locations from taxi trips. We use Dijkstra’s shortest-path algorithm to efficiently calculate where taxis, and we assume other vehicles, likely drove. Since our data set is massive (10 to 15 million taxi trips per month), we used simplified shortest paths and local rerouting.

Problem

We formulate collision prediction as a binary classification problem: “Did a collision occur at this road segment on this day?” However, due to the sparsity of collisions, we chose the more tractable binary classification problem as “How many collisions occurred at this road segment on this day?” Note that we could also frame the classification problem: “Did a collision occur at this road segment on this day?” We then evaluated DSTGCN which was specifically designed for the collision prediction task (Yu et al. 2021). We found DSTGCN performed poorly, perhaps because of their known effectiveness on data sets like ours with a large number of features (Borisov et al. 2022). We then evaluated LightGBM 0.78 ± 0.0005 and XGBoost 0.80 ± 0.0001 while not overly discounting recall for the negative class. We then considered boosting algorithms XGBoost and LightGBM. Both models were strong baselines likely because of their known effectiveness on data sets like ours with a large number of features (Borisov et al. 2022).

Models

We started with standard machine learning models like Logistic Regression, Random Forest, and Gaussian Naive Bayes classifiers. Among these, only the Gaussian Naive Bayes classifier demonstrated non-trivial recall for the positive class. We then considered boosting algorithms XGBoost and LightGBM. Both models were strong baselines likely because of their known effectiveness on data sets like ours with a large number of features (Borisov et al. 2022).

We next evaluated a recurrent GNN (RGNN). We hypothesized that the road structure and traffic patterns interact temporally in the short term and that the recurrent layers could successfully capture these relationships. Unlike Graph WaveNet, the RGNN uses a fixed graph structure which we hypothesize enables it to achieve higher performance on the large network we consider.

Models

We started with standard machine learning models like Logistic Regression, Random Forest, and Gaussian Naive Bayes classifiers. Among these, only the Gaussian Naive Bayes classifier demonstrated non-trivial recall for the positive class. We then considered boosting algorithms XGBoost and LightGBM. Both models were strong baselines likely because of their known effectiveness on data sets like ours with a large number of features (Borisov et al. 2022).

We next evaluated a recurrent GNN (RGNN). We hypothesized that the road structure and traffic patterns interact temporally in the short term and that the recurrent layers could successfully capture these relationships. Unlike Graph WaveNet, the RGNN uses a fixed graph structure which we hypothesize enables it to achieve higher performance on the large network we consider.

Table 1: Results of collision prediction models. Overall support in the test set was 1,803,363 observations: 1,789,838 negative and 13,525 positive examples. The ± denotes standard deviation 10 random seeds. Since the F1-score ignores the imbalanced nature of our data, we use the macro average recall to select the best model.

<table>
<thead>
<tr>
<th>Model</th>
<th>F1-score</th>
<th>Recall (Negative)</th>
<th>Recall (Positive)</th>
<th>Recall (Macro Average)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian NB</td>
<td>0.97±0.0001</td>
<td>0.95±0.0001</td>
<td>0.15±0.0001</td>
<td>0.55±0.0001</td>
</tr>
<tr>
<td>LightGBM</td>
<td>0.78±0.0005</td>
<td>0.64±0.0006</td>
<td>0.80±0.0003</td>
<td>0.72±0.0002</td>
</tr>
<tr>
<td>XGBoost</td>
<td>0.80±0.0001</td>
<td>0.67±0.0001</td>
<td>0.81±0.0001</td>
<td>0.74±0.0001</td>
</tr>
<tr>
<td>DSTGCN (Yu et al. 2021)</td>
<td>0.67±0.2600</td>
<td>0.56±0.2701</td>
<td>0.59±0.1070</td>
<td>0.57±0.0401</td>
</tr>
<tr>
<td>Graph WaveNet (Wu et al. 2019)</td>
<td>0.75±0.0121</td>
<td>0.61±0.0160</td>
<td>0.68±0.0006</td>
<td>0.64±0.0080</td>
</tr>
<tr>
<td>Recurrent GNN (Lite)</td>
<td>0.86±0.0130</td>
<td>0.77±0.0200</td>
<td>0.68±0.0215</td>
<td>0.73±0.0043</td>
</tr>
<tr>
<td>Recurrent GNN</td>
<td>0.87±0.0064</td>
<td>0.78±0.0102</td>
<td>0.74±0.0157</td>
<td>0.76±0.0040</td>
</tr>
</tbody>
</table>

The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)
Table 1 summarizes our findings. Each model was hyperparameter tuned and, when applicable (i.e. in the deep learning setting), trained for 100 iterations. We report on the average performance (plus or minus the standard deviation) over ten random initializations of each model on the same train and test sets.

Figure 3 plots the average predictive effect of the most important features. The number of cars on a road segment and the travel time (in ideal conditions) generally reduce the risk of collisions. We hypothesize this is because driving is cognitively easier in slower conditions (Nilsson et al. 2017). The remaining features we plot all generally increase the risk of collisions. Street length, street width, and speed limit are all associated with higher speed roads which make collisions more likely (Das, Park, and Sarkar 2023). The features with the next largest effects—the radius of curved roads, double level roads, and roads on the border of Manhattan—are all associated with the highways into and out of the island. It is well-documented that locations with speed variation are correlated with more collisions (Li et al. 2013). Finally, bike lanes also have a (small) effect on collisions perhaps because of the dangers of biking in Manhattan (Chen et al. 2012).

**Part 2: Choosing Streets**

In this section, we consider the problem of choosing which road segments to open as a mechanism for reducing traffic and collisions. A natural idea is to choose the road segments with the highest levels of traffic and most collisions. However, there are two issues with this approach:

1. There are complicated endogenous effects. For example, if they are rerouted from road segments with traffic, cars may clog smaller streets or exacerbate gridlock in other heavily congested areas. In addition, more complicated traffic patterns that increase cognitive load can raise the risk of collisions (Engström et al. 2017).

2. There are complicated temporal dynamics so road segments that are beneficial to open in some conditions may be quite harmful to open in others. Days of the week, weather patterns, and special events all impact where and how people drive.

We address these two issues by framing the problem of opening road segments as a reinforcement learning problem. In particular, we incorporate temporal dynamics by considering sequences of days and we incorporate endogenous effects by propagating road openings through time.

Figure 4 shows how we formulate the problem.

**States** States are representations of the city on a given historical day. The representation includes weather and traffic, calculated from actual taxi trips. The representation also includes all infrastructure information but some road segments...
have been hypothetically opened (closed to vehicular traffic). The state carries a list and updates the traffic pattern in each new day as if all road segments on the list were opened (using the approach described below).

**Actions** From a state, the agent selects a road segment to open. Opening a road segment requires rerouting all cars to alternative routes. We accomplish this by finding the top $k$ weighted shortest paths in the network where the road segment is removed and assigning traffic to each path proportional to its total weight. The weight of a road segment is the expected time to travel it in optimal conditions: the product of its posted speed limit and its length.

**Rewards** For each state, we compute the total collision risk and the total car density per lane as a measure of traffic. We compute the density per lane using:

$$\sum_{\text{road segments } \ell} \frac{\text{cars on } \ell \text{ per day}}{\text{traffic lanes on } \ell \times \text{length of } \ell \text{'}}$$

However, the collision risk is much more complicated to compute. The challenge is that the states are hypothetical traffic patterns on real days so we do not actually know how many collisions would have occurred. Our solution is to calculate collision risk using the best model for predicting collisions from the first part of our work. In particular, we compute the predicted collisions from each state and sum the resulting risk of probability along each road segment. We normalize both total collision risk and total traffic using a random day. In order to compute the reward of an action, we use the sum of the collision risk and traffic in the current state minus the same quantity of the next state. Then the reward is positive if and only if the next state has reduced collision risk and traffic. Our general approach is flexible; the sum of collision risk and traffic can easily be reweighted to reflect the priority assigned by domain experts.

The RL agent learns by sampling trajectories (sequences of states, actions, and rewards) to find which road segments are best to open. We investigate 1-month-long trajectories, giving the agent time to observe the long-term effects of opening road segments while also experimenting with different strategies. There are several invalid actions that can prematurely end a trajectory. First, opening a road segment is invalid if there are no cars to reroute (this can happen because we use a single shortest path for inferring taxi trips). Second, opening a road segment is invalid if there is no other directed path from the starting intersection to the ending intersection (this can happen because we limit the road network to Manhattan).

**Local vs. Global Rerouting** When taking an action, we consider local rerouting (instead of global rerouting) because of computational cost. Our model is equivalent to a setting where drivers determine their path and then, along the way, find some road segments are opened and reroute to stay on their chosen path accordingly. Of course, the more realistic setting is that drivers know which road segments are opened and incorporate this information in the path they choose. Unfortunately, because there are tens of millions of taxi trips in our data set each month, we cannot afford to recompute the shortest path for every action. If we did, the average impact than a random selection of streets.

**Q-learning** We solve the RL problem with Q-learning for two reasons: First, Q-learning tends to be more sample-efficient than other RL methods (this is particularly important because both our state space and action space are large) (Jin et al. 2018). Second, Q-learning produces a value which we can interpret as the expected long-term reward of opening a road segment. Then the road segment with the largest Q-value corresponds to the best one to open while accounting for endogenous effects and temporal dynamics.

**Experiments**

**Data** Collision data comes from *NYC Open Data*, which releases cleaned police reports (OpenData 2022). Infrastructure data is a road-bed representation of NYC and contains about a hundred features like road type, traffic direction, and other features for each road segment (Planning 2022). Weather data comes from NOAA (NOAA 2022), and is geographically coarse (only from a single weather station in Central Park) but updated hourly. Finally, our taxi trip data comes from NYC’s Taxi and Limousine Commission. Each trip contains features like the start and end GPS coordinates, trip duration, and time of trip. Because taxi trips stopped being shared with exact start and end locations in 2016, we conducted our experiments on data from 2013, 2014, and 2015.

**Evaluation** We used implementations of Gaussian NB, XGBoost and LightGBM for collision prediction model baselines. We used (and modified) existing implementations of DSTGCN and Graph WaveNet (Yu et al. 2021; Wu et al. 2019). We implemented our recurrent GNN models using Pytorch (Paszke et al. 2019). We hyperparameter-tuned with
Q-learning provides better long term reward than the open streets initiative. The middle plot of Figure 6 shows the 121 streets selected by the open streets initiative in yellow and the 121 streets with the highest Q-value in blue. The streets in the open streets initiative (yellow) are concentrated in Downtown Manhattan and completely absent from Midtown and most of the East Side. The right plot of Figure 6 confirms the discrepancy. Most neighborhoods have more streets with large Q-values (light and dark blue) while a handful have many more streets in the open streets initiative (dark yellow). From these two plots of Manhattan, we find that the streets with the highest Q-value are more geographically diverse. A concern with the open streets initiative is inequity in the neighborhoods that benefit from the program (Hazari 2021). We believe an advantage to using a principled approach is a more equitable geographic distribution.

Most of the worst streets to open are East-West. In the left plot of Figure 6, almost all the streets with the lowest Q-values (dark red) are East-West. Since Manhattan is optimized for North-South travel with large one-way avenues and synchronized traffic signals, we believe the low Q-values for East-West streets are an emergent property of our model (Owen 2004).

Limitations and Future Work

Due to the size of our data (tens of millions of taxi trips per month and 19,391 road segments), we reroute individual taxi trips around opened streets. This corresponds to the setting where a driver gets to a road and only then learns it is opened, while the more realistic setting corresponds to a driver planning their route with prior knowledge of opened road segments. Future work could use more compute (or a better technique) to realistically reroute traffic after streets are opened. We discuss the effective resistance as one such possible technique in the extended version online.

We intentionally focused on NYC to demonstrate a proof-of-concept and integrate feedback from local transportation experts. We leave applying the approach, and even the networks we trained, to other cities as future work. We consider the objectives of reducing traffic and collisions. However, there are more objectives such as pedestrian utilization or tourist interest that could make streets desirable to open. Future work could integrate other objectives by augmenting the reinforcement learning reward function. Because collision data is necessarily sparse, prior work has used cameras and sensors to detect near-collision events (Wang and Chan 2017; Osman et al. 2019). Future work could use such data to improve the modeling; however, to the best of our knowledge, the requisite number of cameras and sensors are not available for even a fraction of the segments in Manhattan.

Neural networks are notoriously difficult to interpret (Zhang et al. 2021b). This is especially a problem in the high stakes domains of road networks that we applied them to. We used the integrated gradients method to analyze the feature importance our RGNN but we believe additional interpretability work would benefit models for predicting collisions and opening streets.
Acknowledgements
RTW and LR were supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-2234660. We thank Suzana Duran Bernardes for helpful discussions and direction to data sources.

References


Planning, N. 2022. LION Single Line Street Base Map. NYC Department of Planning.


Zhao, H.; Cheng, H.; Mao, T.; and He, C. 2019. Research on Traffic Accident Prediction Model Based on Convolutional Neural Networks in VANET. 2019 2nd International Conference on Artificial Intelligence and Big Data (ICAIBD).