SMARTER PARKING: USING AI TO IDENTIFY PARKING INEFFICIENCIES IN VANCOUVER

Devon Graham
Department of Computer Science
University of British Columbia
Vancouver, BC
drgraham@cs.ubc.ca

Satish Kumar Sarraf
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, BC
ssatish@ece.ubc.ca

Taylor Lundy
Department of Computer Science
University of British Columbia
Vancouver, BC
tlundy@cs.ubc.ca

Ali MohammadMehr
Department of Computer Science
University of British Columbia
Vancouver, BC
alimm@cs.ubc.ca

Sara Uppal
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, BC
sarau@ece.ubc.ca

Tae Yoon Lee
Department of Statistics
University of British Columbia
Vancouver, BC
taeyoon.lee@alumni.ubc.ca

Hedayat Zarkoob
Department of Computer Science
University of British Columbia
Vancouver, BC
hzarkoob@cs.ubc.ca

Scott Duke Kominers
Harvard Business School
Boston, MA
kominers@fas.harvard.edu

Kevin Leyton-Brown
Department of Computer Science
University of British Columbia
Vancouver, BC
kevinlb@cs.ubc.ca

March 24, 2020

ABSTRACT

On-street parking is convenient, but has many disadvantages: on-street spots come at the expense of other road uses such as traffic lanes, transit lanes, bike lanes, or parklets; drivers looking for parking contribute substantially to traffic congestion and hence to greenhouse gas emissions; safety is reduced both due to the fact that drivers looking for spots are more distracted than other road users and that people exiting parked cars pose a risk to cyclists. These social costs may not be worth paying when off-street parking lots are nearby and have surplus capacity. To see where this might be true in downtown Vancouver, we used artificial intelligence techniques to estimate the amount of time it would take drivers to both park on and off street for destinations throughout the city. For on-street parking, we developed (1) a deep-learning model of block-by-block parking availability based on data from parking meters and audits and (2) a computational simulation of drivers searching for an on-street spot. For off-street parking, we developed a computational simulation of the time it would take drivers drive from their original destination to the nearest city-owned off-street lot and then to queue for a spot based on traffic and lot occupancy data. Finally, in both cases we also computed the time it would take the driver to walk from their parking spot to their original destination. We compared these time estimates for destinations in each block of Vancouver's downtown core and each hour of the day. We found many areas where off street would actually save drivers time over searching the streets for a spot, and many more where the time cost for parking off street was small.

The identification of such areas provides an opportunity for the city to repurpose valuable curbside space for community-friendly uses more in line with its transportation goals.

***K*eywords** Parking · Street Parking · Deep Neural Networks · Occupancy Prediction · Parking Simulation

1 Introduction

In many urban areas, trying to find curbside parking can be an unpleasant experience. In peak hours, available spots are rare and drivers can cruise from block to block without finding one. Even having found a spot, drivers may then need to walk a significant distance back to their destination. This takes time and clogs up city streets with unnecessary congestion. In fact, in a well-known study, Shoup found that on average 30% of traffic on city streets can be the result of such cruising (Shoup, 2006). This excessive congestion has many disadvantages. Drivers searching for parking are distracted, reducing safety for pedestrians and cyclists (Edquist *et al.*, 2011). With more cars on the road, travel times are increased for everyone, including public transit vehicles. More slow-moving cars on roads means increased emissions (Rouphail *et al.*, 2001). Thus, it is important for modern cities to manage their parking resources efficiently.

As an alternative to curbside parking, drivers have the option to park in off-street lots (by which we refer equally to parkades, parking garages, and ground-level lots). Such lots often offer many more available spots than on-street parking. However, parking off-street has its own drawbacks. Drivers need to first find the lot, find a free spot, pay, and then walk back to their destination. It is therefore not immediately clear whether most drivers save time by seeking on- or off-street parking.

Previous studies have examined the trade-off between searching for curbside parking and parking in the nearest lot, most through the lens of examining pricing schemes (Ottosson *et al.*, 2013; Shoup, 2006). In particular, Shoup looked at how drivers might think about on- and off-street parking by assuming that curbside parking is free but rare while off-street spots are always available but expensive (Shoup, 2006). He observed that if the time it takes for drivers to cruise for free curbside parking exceeds a certain threshold, they will prefer paying for off-street parking to save time. However, price is not the only factor affecting drivers' decisions (Yan *et al.*, 2019). Furthermore, drivers trade time for money differently depending on their situations. If a driver has an important meeting with a client, she will gladly pay more for parking to avoid being late and risking losing the client; if a driver holds an expensive concert ticket, paying a high price for parking is not as bad as missing part of the show.

Our work approaches the problem of curbside vs. lot parking from another angle: we consider only the difference in time between the two alternatives, and think of price as a tool that can be used to change driver behaviour. The off-street lots we consider are all owned by the City of Vancouver—as, of course, are the on-street spots—meaning that adjusting prices is within the power of the City's decision makers. We show how cutting-edge AI tools can be leveraged along with real-world data to build realistic models of on- and off-street parking procedures. We show that despite the costs of off-street parking mentioned above, drivers can sometimes save time by parking in off-street lots rather than cruising in search of on-street parking.

In order to simulate cruising for parking, we predict curbside parking occupancy from observed payment data. Various methods exist for making such predictions (Rajabioun and Ioannou, 2015; Tamrazian *et al.*, 2015; Yang and Qian, 2017; Yang *et al.*, 2019), but none of these methods works with the sort of highly incomplete data we encountered in practice. Aggregated payment data is only weakly predictive of actual occupancy rates (Ottosson *et al.*, 2013): illegally parked cars still consume a spot, and paid drivers can leave both before and after their meter expires. While curbside payments are performed via on-street meters that accept credit cards, cash, and app payments, we only had access to app payments. We were able to calibrate these noisy observations with a very limited number of occupancy checks performed by a parking inspector. We show that in this setting it is nevertheless possible to build accurate models using deep neural networks.

The remainder of this paper begins with a brief literature review in Section 2, contrasting approaches taken in the literature with our own methods. Then, we describe the methods we developed to simulate both on- and off-street parking in Section 3. Section 4 describes the data used in our simulations and Section 5 analyzes the results. Finally, we conclude and discuss some future directions in Section 6.

2 Literature Review

Others have previously considered methods for simulating the parking search procedure, as well as comparing on- and off-street parking, and predicting occupancy from payment data. We review some relevant work below, highlighting methodological differences in the approaches proposed in this paper.

2.1 On-Street Parking Simulators

Mannini *et al.* (2017) performed an influential study of on-street parking time estimation, computing on-street parking search time by using Floating Car Data obtained from probe vehicles. Belloche (2015) used survey-based data to compute parking search time based on an off-street parking search time model proposed by Axhausen *et al.* (1994). Most related to our own work, Dowling *et al.* (2019) viewed streets as a network, representing streets and intersections as edges and nodes in a graph. Then, using the transaction data to estimate true occupancy, Dowling *et al.* (2019) calculated various network-level measures such as congestion. Our work uses transaction data to estimate true occupancy using a deep learning model and estimates the parking search time with a simple search algorithm using Monte-Carlo simulation.

2.2 Off-Street Parking Simulators

While there exist agent-based choice models for parking in general (Vuurstaek *et al.*, 2018; Benenson *et al.*, 2008; Waraich and Axhausen, 2012; Bischoff and Nagel, 2017) and survey-based studies for off-street parking search time (Axhausen *et al.*, 1994; Teng *et al.*, 2002), literature on explicitly modeling parking in an off-street parking lot is quite limited. We propose a simple simulation-based approach to compute the time it takes for a car to park in an off-street parking lot.

2.3 On-Street vs. Off-Street Parking

There has been active research in parking policy and regulation for both on-street parking (Marshall *et al.*, 2008; Biswas *et al.*, 2017) and off-street parking (Barter, 2010). On-street parking is evaluated on a combination of three areas: safety for drivers, cyclists, and pedestrians; road congestion; and economic activity (Marshall *et al.*, 2008). Off-street parking lots serve as extra supply to accommodate excess on-street parking demand. Studies have suggested using pricing to manage the markets for on-street and off-street parking individually (Ottosson *et al.*, 2013). However, these studies do not offer a collective assessment of on-street and off-street parking. Inci and Lindsey (2015) considers the relationship between on-street parking and off-street parking by examining price. Our own work uses time as a common metric to compare on- and off-street parking and identify destinations for which it takes less time to park off-street than on-street.

2.4 Parking Occupancy Prediction from Transaction Data

The most straightforward method of estimating parking occupancy from transaction data is simply to assume that every driver leaves exactly once her paid session expires and that all drivers pay for parking. Fiez and Ratliff (2017) and Dowling *et al.* (2019) estimate parking occupancy in this way at a block-face level over the course of an hour. They acknowledge the limitations of this method, noting particularly that drivers tend to leave before their paid parking time, leading to over-estimation of occupancy. In recent years, statistical and machine learning models have been developed to deal with such issues and predict parking occupancy more accurately using additional data. These include multivariate spatial-temporal models (Rajabioun and Ioannou, 2015), unsupervised clustering (k-means and k-nearest neighbours) (Tamrazian *et al.*, 2015), probabilistic model (Yang and Qian, 2017), and most notably a series of deep learning approaches (Yang *et al.*, 2019; Zheng *et al.*, 2015; Kepaptsoglou *et al.*, 2014). These deep learning approaches rely heavily on the availability of actual occupancy data from sensors and surveys, which indeed is becoming increasingly available. Both Yang *et al.* (2019) and Kepaptsoglou *et al.* (2014) leveraged a model called recurrent neural nets to predict occupancy rates from complete transaction data. Another approach by Zheng *et al.* (2015) used the data from SFPark (approximately 2 million block-level samples) to construct a deep neural network, comparing performance both to regression trees and support vector machines. In contrast to all of this work, our training data is sparse in two senses. First, our transaction data is not complete: we have access only to a single mode of payment, which comprises about 60% of total meter transactions. Second, the observations used for our ground truth occupancy levels are based on human surveys rather than sensors, giving us considerably less data than needed by the approaches just mentioned; furthermore, this data is not uniformly sampled across different times of day. This paper proposes an alternate deep learning approach for estimating and predicting parking occupancy by combining incomplete transaction data from parking meters with limited in-person counts.

3 Methods

3.1 On-Street

We construct a simulation of the time it takes to park on-street by modeling the behaviour of a driver searching for a spot under the assumption that the driver starts at his/her destination. To generate accurate search times, we first need

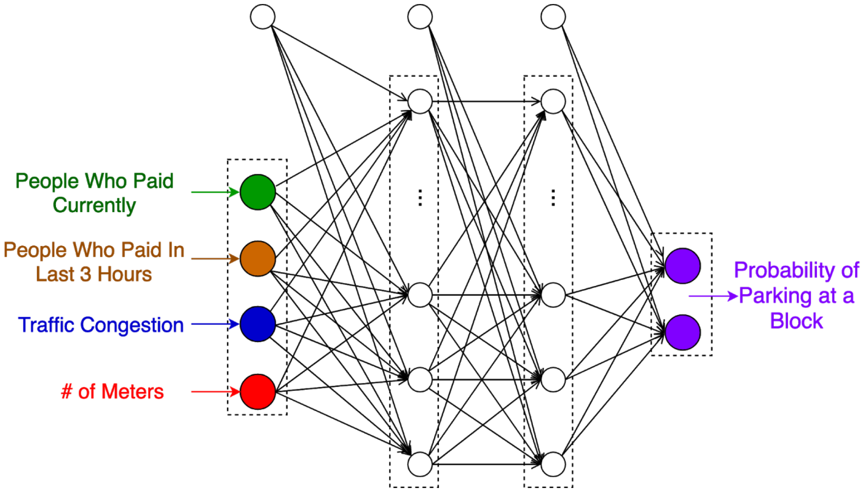
People Who Paid Currently
People Who Paid In Last 3 Hours
Traffic Congestion
# of Meters
Probability of Parking at a Block

Figure 1: *Neural network design used to compute the probability of parking at a block.*

predictions of the availability of parking at each individual block of the city, for every hour of the day. To generate these predictions we use a deep neural network trained to predict occupancy from payment data. In this section, we first give some brief background on feed-forward neural networks. We then describe the specific network we construct to solve this problem, and finally, we describe how we use these predictions to simulate search times for on-street parking.

3.1.1 Neural Networks

Neural networks are a broad class of functions that map inputs to outputs in a complex fashion. These networks are built from a sequence of $k \in \mathbb{N}$ "layers", where the output of each layer is fed as the input to the next. Each layer consists of a non-linear function σ applied to a linear transformation of its input. The linear transformation in each layer i is parameterized by a weight matrix, W_i, and a bias vector, b_i. Thus, a neural network is just a function of the form

$$f(x) = W_k\ \sigma\Big(W_{k-1}\ \sigma\Big(\ \dots\ \sigma(W_2\ \sigma(W_1 x + b_1) + b_2)\dots\Big) + b_{k-1}\Big) + b_k, \tag{1}$$

where x is the input. Neural networks can approximate any function to arbitrary precision, if the matrices W are chosen to be large enough. Given inputs x and target values y, training the network means finding values of the parameters W and b for each layer such that the network's output, $f(x)$ is close to the true, known output y.

3.1.2 Occupancy Prediction Network

We assume that we are given a partial record of payments made at parking meters in each block face in the city. Inferring occupancy from payment information is not as straightforward as it may seem: drivers may leave before their time is up, in which case payment data overestimates occupancy; alternately, they may overstay or even park without paying at all, in which case payment data will underestimate occupancy. Our goal is to predict the true occupancy of a block as a function of observed payment data for that block.

For each block face and time, our model is given two input values which are derived from the payment data. We augment this with two input values that do not directly depend on the payment data, but relate to the block face in question. The input values are represented as a vector of length 4 consisting of the following values:

1. The total number of paid parking sessions in the block at the given time;
2. The total number of paid parking sessions in the block over the 3 hour period before the given time ("popularity")
3. The length of the block in meters; and
4. The predicted time to drive the length of the block divided by the length of the block ("congestion").

For each block and time, the neural network outputs a vector of length two, whose elements are forced to sum to 1 and are thus interpretable as probabilities. To train the network, we minimize the cross-entropy loss between these probabilities and the true, observed occupancy state of the block. In this way, the network learns to approximate the true probability of parking being available in a block at a given time, as a function of the given data. Using the notation of Section 3.1.1, our network consists of an input layer, two hidden layers and an output layer (thus $k = 3$; the input layer does not count as a layer). The connections between these layers are weight matrices of sizes 4×30, 30×30, and 30×2 respectively. Including bias vectors, this fully-connected neural network is described by $1,142$ parameters. As a non-linear function we use ReLU: $\sigma(x) = \max(0, x)$. We train the network on ten different random training and validation splits, with the validation set always constituting 20% of the total data, and iterating over the training data 150 times for each split.

Once the network is trained, we predict the existence of an empty meter at a given time (and thus whether a driver can park) in any block by simply feeding the model the payment and block-level features. Since the model has been trained to reproduce the probability that there will be an available spot, we can use this output directly in our simulation. To evaluate our network's performance, we compare its results to a logistic regression model as baseline that is given the same input variables our network is given. We evaluate both the network and the regression in terms of accuracy (with threshold 0.5) and cross-entropy.

3.1.3 On-street Search Simulator

A key portion of our simulator models a driver's search for on-street parking, based on noisy observations of traffic meter usage. We simulate search time as the sum of three factors: the minimum search time, the estimated time required to drive on nearby streets while searching, and the estimated time required to walk back to the destination.

More specifically, we constructed a representation of downtown Vancouver using OSMnx (Boeing, 2017), a Python package for modeling road networks. The city is modeled as a graph structure: intersections correspond to "nodes" and street segments correspond to "edges" connecting pairs of nodes. We assume that drivers begin searching for parking in the block that contains their destination. Using the probabilities predicted by the neural network, we model the on-street parking search process for each block face. During the search process, for any edge the vehicle traverses, we say it can park on that block with the probability predicted by the neural network. If it parks, we end the search process. We assume that on-street parking requires some minimum amount of time, $t_{\min}$; this corresponds to the time required for parallel parking and paying at the meter. We set $t_{\min} = 210$ seconds, following Shoup (2006). If the vehicle fails to park, it moves to the next edge and tries again. Once the vehicle is parked, the driver walks from the parking spot to the original destination; we assume both locations are in the middles of their respective blocks. We obtained driving and walking times from the Google Maps Directions API, assuming that walking times remain constant throughout the day. For a single vehicle, let $d_1, d_2, ..., d_n$ be the driving times for the blocks the vehicle traverses, and $w_1, w_2, ..., w_m$ be the walking times for blocks the driver traverses as a pedestrian. Then the total on-street search time, T, is modeled as

$$T = t_{\min} + \left(\frac{d_1}{2} + \sum_{i=2}^{n} d_i - \frac{d_n}{2} \right) + \left(\frac{w_1}{2} + \sum_{j=2}^{m} w_j - \frac{w_m}{2} \right). \tag{2}$$

With each block face and for each time of the day, we use the simulator to produce many samples of the value T, and then average to estimate mean search time.

A key consideration in simulating driver search is reasoning about how drivers choose which block to search next. We model this as a random process, where the probability of the driver choosing a block, i, depends on four quantities:

1. D_i, the distance between block i and the driver's destination (drivers are more likely to turn back towards the destination than away, making blocks further away less likely to be selected);
2. N_i, the number of times the driver has already checked block i during their search (each repetition makes a block less likely to be checked);
3. E_i, the time that has elapsed since the driver last checked this block (the more recent, the less likely the block is to be rechecked); and
4. P_i, the probability of parking being available on block i, as predicted by the neural network model (blocks with more available parking are more likely to be visited). Note that this only incorporates the parking probabilities adjacent to the driver's current intersection. This is because we assume that drivers do not have accurate predictions about parking probabilities across the whole city, but that drivers can visually inspect blocks adjacent to the intersection to obtain about as much information as given by the predictive model.

We form a linear combination of these quantities,

$$Z_i = w_D D_i + w_N N_i + w_E E_i + w_P \frac{1}{P_i},$$

where we hand tuned the w parameters to produce results we considered realistic, setting them to $w_D = -1, w_N = -15, w_E = 15, w_P = -1$.

Observe that because the quantity Z_i could be greater than 1, or even negative, we cannot interpret it as a probability. Instead, we interpret it as the log-odds of choosing this block, and convert it to a probability with the softmax function

$$\Pr[\text{choose Block i}] = \frac{e^{Z_i}}{\sum_j e^{Z_j}}.$$

This produces a realistic approximation of a driver looking for parking since it (i) discourages drivers from travelling too far from their destinations; (ii) prevents drivers from re-checking the same block over and over; and (iii) incorporates knowledge of occupancy levels on the blocks the driver could reasonably see at the time of their decision.

3.2 Off-Street

We estimate the time it takes to park in the closest off-street lot for each block face in the city. To do so, we compute the expected times for a driver (1) to drive from a given destination to that lot; to search for a spot in the lot and park; (3) to walk back to their destination. To compute (1) and (3)—taking into account likely traffic at a given time of day—we appeal to historical traffic data via the Google Maps Directions API. To estimate (2), the time spent searching in the lot, we use a Monte Carlo simulation, described next.

3.2.1 Time Spent Searching in Lots

Our simulation of the time spent in a lot is based on entry and exit data collected from several city-owned lots. For simplification, we assume that all lots are one-dimensional manifolds: i.e., that they have a single entrance and exit with no branching paths. We assume parking in a lot requires some minimum amount of time t'_{min}. In practice we set $t'_{\text{min}} = 60$ seconds. (This may be an overestimate, but if so this only serves to help our argument that lots can often be better for drivers than on-street parking.) We model the time a driver spends in the lot as influenced not only by the number of vehicles currently in the lot, but also by the number of other vehicles entering and exiting the lot at the same time.

We consider a small unit of time (in practice, 60 seconds) and model the number of cars to arrive and to depart in that unit of time as Poisson distributed random variables with means λ_a and λ_d, respectively. The parameters λ_a and λ_d are determined from the observed entry and exit data for each lot and each hour of the day and for each day of the week, averaged over 12 weeks. We keep track of the state of the lot by updating each spot in each unit of time. When it is time for a vehicle to depart, we select that vehicle uniformly at random among the lot's occupied spots.

We assume that vehicles park in the same order that they arrive, and that each one parks in the first available spot it finds. We assume that drivers will wait 30 seconds for a departing car and then take that spot if necessary. Vehicles need to wait for those in front of them to park. The k^{th} vehicle to arrive may wait for up to k departing vehicles to vacate stalls, or may wait for none, so we assume that on average it waits for half of them. While driving in the lot, we use a constant time t_1 as the time it takes to drive past a single parking stall.

Because the minimum time t'_{min} includes paying for parking, and unlike pulling into spots, not all vehicles can always pay at the same time, we assume that the k^{th} vehicle must also wait 1/2 of the minimum time for the vehicle in front of it, 1/4 of the minimum time of the second vehicle in front of it, 1/8 of the minimum time of the third vehicle, etc.

Then, if $N_a \sim_d Poisson(\lambda_a)$ vehicles arrive and $N_d \sim_d Poisson(\lambda_d)$ vehicles depart in this small period of time, the k^{th} vehicle to arrive waits for T_k seconds, where

$$T_k = t'_{\text{min}} + S_k t_1 + \frac{\min(k, N_d)}{2} t_{\text{wait}} + \sum_{i=1}^{k-1} \frac{1}{2^i} t_{\text{min}} \tag{3}$$

seconds, where S_k is the spot that the k^{th} driver parks in according to the simulation, updated based on N_a. For each $k = 1, ..., N_a$, we consider T_k as a single Monte Carlo sample, repeat the process 20 times, and average over all samples to determine the total average search time for each lot and each time of day. We used a value of $t_1 = 0.54$ seconds, based on data recording parking activity in a large local lot, averaging the time it takes to drive through the whole lot (while following the posted speed limit) over the lot's total number of stalls.

4 Data

4.1 Geographic Data

The city of Vancouver offers many publicly available data sets in its *Open Data Catalogue* (`https://data.vancouver.ca/datacatalogue`). We made use of two of these: *Parking Meter Data* contains the location, rates and time limits for approximately 10,000 parking meters in the city, and *City Streets* associates each meter in the Parking Meter dataset with its corresponding block face.

4.2 Occupancy Surveys

To train our neural network for predicting on-street parking spot occupancy, we used (non-publicly available) data from occupancy surveys done by the City of Vancouver on each of the meters on 372 block faces in Vancouver's downtown area. The survey data consisted of timestamped, meter-level surveys. However, since the meters in each block face were surveyed at nearly the same time, we combined all surveys of the meters in a block face that occurred in the same 30-minute period to obtain a single snapshot of the block face. Overall, each block face was checked on average 8 times, with 4 checks in the morning and 4 checks in the afternoon; after combining meter-level data, we had a total of 3084 block-level occupancy check samples. However, time stamps were missing for 17% of the meter-level occupancy checks, making those checks unusable. These were distributed roughly evenly across meters, so we had 7 usable samples for most meters.

4.3 Google Maps Data

The Google Maps Directions API can be used to estimate travel time between any two points in a city that are connected by roads. Ideally we would simply tell Google Maps the route to drive and report the time to drive it. However, the number of such calls that can be made to the API is limited. Instead, we used the API to find driving and walking times for each block face in the city by getting the travel time between the centers of the intersections at either end of the block. We then reconstructed travel time between pairs of points in the city by labeling the road graph with these values and directly running a shortest path algorithm (e.g., Dijkstra's algorithm).

4.4 City-Owned Lot Data

We were given access to proprietary parking data from 21 city-owned lots, containing between 9 and 643 stalls. For each lot we used hourly data from a 3 month period. In each hour the data recorded the number of entries and the amount of time each vehicle paid for. Unfortunately, this did not give us an exact record of when each vehicle left the lot. This problem was particularly pronounced when combined with promotional rates (e.g., a flat rate to park until 6:00 PM). If we naively treated every vehicle as leaving at the moment its parking expired, we would have seen huge artificial spikes at times that represent rate boundaries. To deal with this, we applied a Gaussian smoothing procedure to the departure data for each of these peaks. We then redistributed the departures in the peaks back 12 hours, to account for vehicles leaving the lot at different times, with the quantities distributed to each hour proportional to the probabilities of a left-half-Gaussian distribution with mean at the peak and standard deviation 3.5.

5 Results

5.1 Occupancy Prediction Network

To evaluate the performance of the occupancy prediction neural network, we compare it with a logistic regression model as baseline using cross entropy loss on the validation set. Cross entropy is a measure of difference between two probability distributions; a model with low cross entropy loss does a good job of approximating the true distribution of the data. Under this measure, our network on average outperformed the baseline by 9%: the network achieved average validation cross entropy loss of 0.51 compared to 0.56 for the baseline. We note however that our network was relatively similar to the baseline in terms of validation accuracy, achieving 79% vs 77%. In this case we consider cross entropy loss to be a more meaningful measure of performance, because our model's core purpose is to predict a probability (the probability that a vehicle can park in a given block face).

5.2 Parking Inefficiencies

Using the search times generated by our two simulators we estimated the time it would take to park both on- and off-street when driving to a destination in each block in the city. Figures 2 and 3 show the total estimated time for

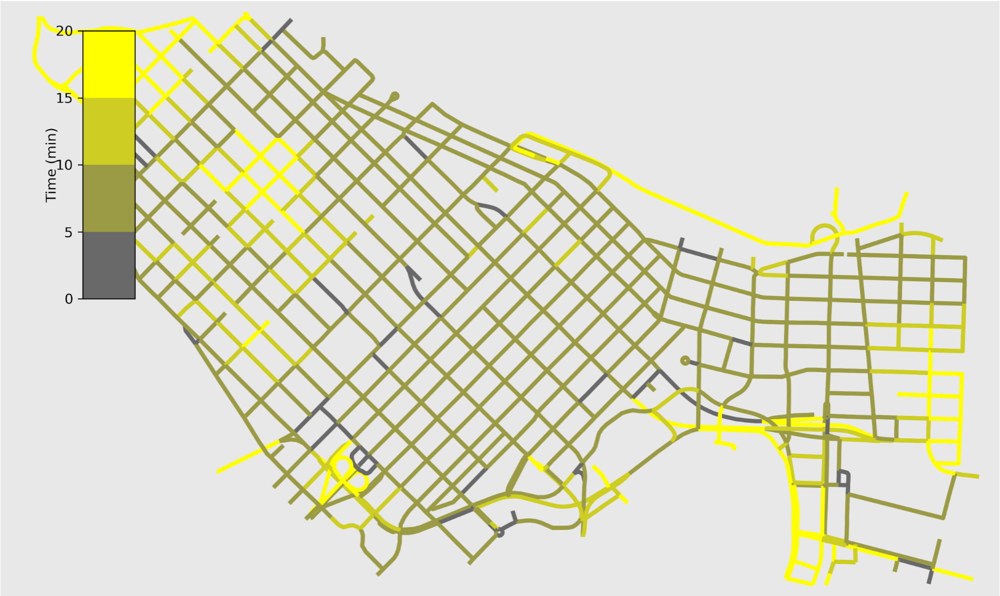

Figure 2: *On Street: Total estimated time to park on-street (including search, park, and walking times) with each block as the assumed destination of the driver. Yellow indicates it will take a relatively long time to park on-street if this block is the driver's final destination. Search times tend to be higher in areas with fewer meters. The map shown is for 12:00-1:00pm on a Friday.*

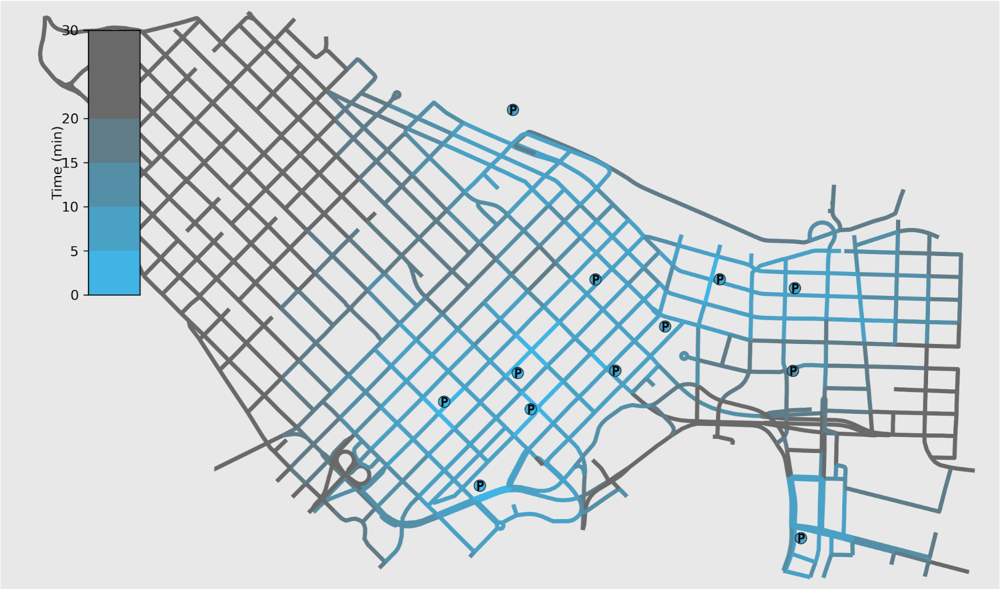

Figure 3: *Off Street: Total estimated time to park in an off-street lot (including driving and walking time, and searching in the lot) with each block as the assumed destination of the driver. Off-street lots are indicated with P icons. Blue indicates a relatively short time to park in a nearby lot if this block is the driver's final destination. Search times tend to be lower in areas nearer to lots. The map shown is for 12:00-1:00pm on a Friday.*

parking on-street and off-street, respectively. Not surprisingly, the on-street map (Figure 2) shows it will take a relatively long time to park on-street in the west end of the city and around the Granville St. area, where meters are less common. Similarly, the off-street map shows that parking off-street is quickest when a driver's destination is near an off-street lot.

It is most interesting to examine the difference between these two maps. Figure 4 shows how much time a driver could *save* by parking off-street, with each block as the assumed destination. Blocks coloured green indicate areas that are at

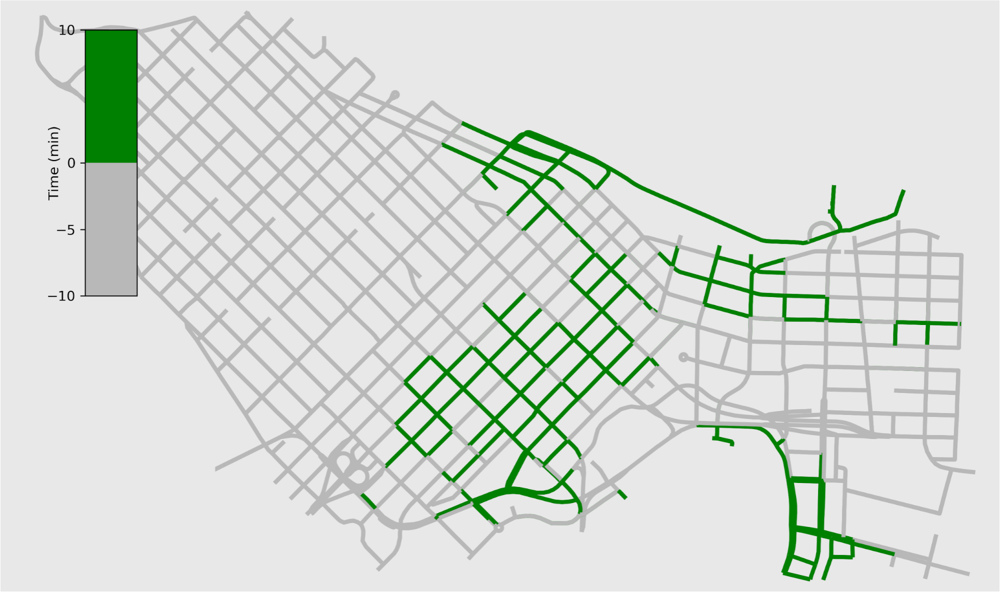

Figure 4: *Areas of Opportunity: Difference in estimated time between parking off-street and parking on-street. Blocks coloured green indicate drivers can actually save time by parking off-street rather than searching for curbside spots if these blocks are their destinations. Curbside parking in these areas can be eliminated, with nearby lots absorbing the excess capacity. This space can then serve more community-friendly purposes. The map shown is for 12:00-1:00pm on a Friday.*

least as accessible from lots as they are from on-street meters. Most of these areas can be reached about 5 minutes *faster* by parking off-street, with some being as much as 10 to 15 minutes faster. In some cases the issue is simply a lack of parking meters on the destination street, which means that drivers must walk some distance regardless of their parking modality. In other areas (e.g., around Granville St., Yaletown, and Chinatown, seen in the center-right of Figure 4) congested roads overlap with areas that are easily accessible from off-street lots. In these areas searching for on-street parking can take a long time due to the congestion, but the presence of nearby lots means off-street parking is relatively quick. These areas represent an opportunity for the city to effect change in the way people use parking resources in the city.

6 Conclusion

While on-street parking has certain advantages, its overuse has negative side effects that impact everyone in the city. Drivers who are looking for parking occupy valuable road space unnecessarily and also tend to be distracted. Parked cars can obscure pedestrians and cyclists from drivers' sight. At the same time, many off-street parking lots have large amounts of excess capacity at times of day when on-street parking is in high demand.

We examined parking usage on metered blocks and in city-owned, off-street lots in downtown Vancouver. We used deep learning to predict true on-street occupancy rates from partial payment and other, block-level data. We used Monte Carlo simulation in conjunction with Google Maps driving and walking times to model the amount of time taken to park both on- and off-street, with each block face in the city as the assumed destination. We identified areas where it would be at least as fast for drivers to park in nearby lots as to search the area and then park on-street.

Since the particular lots we considered are owned by the City, the City has the power to change the way they are operated. Thus, these areas represent a clear opportunity to improve the city as a whole. Curbside space is a valuable resource that modern municipalities can utilize to improve the lives of all their residents. In particular, on-street parking reduces safety for drivers, pedestrians and cyclists alike by obscuring vision and increasing the number of distracted drivers on the road. What's more, curbside lanes can be converted to dedicated bike or transit lanes, helping the City of Vancouver maintain its vehicle mode-share targets. Alternatively, these lanes can be converted to parklets or green spaces, making these areas more community-friendly.

How then can drivers be incentivized to use lots instead of curbside space? One simple way is through better information, either in the form of signage or through a website or app showing off street availability. More overt incentives are also feasible. For the most part, off-street parking rates in Vancouver are higher than nearby on-street parking rates. Better aligning pricing and time restrictions between on- and off-street parking could encourage more drivers to move off-street. Most ambitiously, a reservation system could provide the strongest incentives by enabling fixed pricing and eliminating uncertainty for drivers, allowing them to head directly to a lot from their home. Any of these measures could help the City make better use of valuable on-street space at little to no cost to drivers.

References

Kay W Axhausen, JW Polak, M Boltze, and J Puzicha. Effectiveness of the parking guidance information system in frankfurt am main. *Traffic Engineering+ Control*, 35(5):304–9, 1994.

Paul A Barter. Off-street parking policy without parking requirements: a need for market fostering and regulation. *Transport Reviews*, 30(5):571–588, 2010.

Sylvain Belloche. On-street parking search time modelling and validation with survey-based data. *Transportation Research Procedia*, 6:313–324, 2015.

Itzhak Benenson, Karel Martens, and Slava Birfir. Parkagent: An agent-based model of parking in the city. *Computers, Environment and Urban Systems*, 32(6):431–439, 2008.

Joschka Bischoff and Kai Nagel. Integrating explicit parking search into a transport simulation. *Procedia Computer Science*, 109:881–886, 2017.

Subhadip Biswas, Satish Chandra, and Indrajit Ghosh. Effects of on-street parking in urban context: a critical review. *Transportation in developing economies*, 3(1):10, 2017.

Geoff Boeing. Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers Environment and Urban Systems*, 65:126–139, 07 2017.

Chase P Dowling, Lillian J Ratliff, and Baosen Zhang. Modeling curbside parking as a network of finite capacity queues. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

Jessica Edquist, Christina M. Rudin-Brown, and Michale G Lenné. The effects of on-street parking and road environment visual complexity on travel speed and reaction time. *Accident Analysis and Prevention*, 2011.

Tanner Fiez and Lillian Ratliff. Data-driven spatio-temporal analysis of curbside parking demand: A case-study in seattle. *arXiv preprint arXiv:1712.01263*, 2017.

Eren Inci and Robin Lindsey. Garage and curbside parking competition with search congestion. *Regional Science and Urban Economics*, 54:49–59, 2015.

Konstantinos Kepaptsoglou, Matthew G. Karlaftis, and Eleni I. Vlahogianni. Exploiting new sensor technologies for real-time parking prediction in urban areas. 2014.

L Mannini, E Cipriani, U Crisalli, A Gemma, and G Vaccaro. On-street parking search time estimation using fcd data. *Transportation Research Procedia*, 27:929–936, 2017.

Wesley E Marshall, Norman W Garrick, and Gilbert Hansen. Reassessing on-street parking. *Transportation Research Record*, 2046(1):45–52, 2008.

Dadi Baldur Ottosson, Cynthia Chen, Tingting Wang, and Haiyun Lin. The sensitivity of on-street parking demand in response to price changes: A case study in seattle, wa. *Transport Policy*, 25(C):222–232, 2013.

Tooraj Rajabioun and Petros A Ioannou. On-street and off-street parking availability prediction using multivariate spatiotemporal models. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2913–2924, 2015.

Nagui M. Rouphail, H. Christopher Frey, James D. Coylar, and Alper Unal. Vehicle emissions and traffic measures: exploratory analysis of field observations at signalized arterials. *Annual Meeting of the Transportation Research Board*, 2001.

Donald C Shoup. Cruising for parking. *Transport Policy*, 13(6):479–486, 2006.

Arbi Tamrazian, Zhen Qian, and Ram Rajagopal. Where is my parking spot? online and offline prediction of time-varying parking occupancy. *Transportation Research Record*, 2489(1):77–85, 2015.

Hualiang Teng, Yi Qi, and Ping Yi. Parking search time and information identification for off-street spaces in new york city. *Transportation research record*, 1800(1):44–52, 2002.

Jan Vuurstaek, Luk Knapen, Bruno Kochan, Tom Bellemans, and Davy Janssens. First steps towards a state-of-the-art parking simulator. Elsevier, 2018.

Rashid A Waraich and Kay W Axhausen. Agent-based parking choice model. *Transportation Research Record*, 2319(1):39–46, 2012.

Xiang Yan, Jonathan Levine, and Robert Marans. The effectiveness of parking policies to reduce parking demand pressure and car use. *Transport Policy*, 73:41–50, 2019.

Shuguan Yang and Zhen Sean Qian. Turning meter transactions data into occupancy and payment behavioral information for on-street parking. *Transportation Research Part C: Emerging Technologies*, 78:165–182, 2017.

Shuguan Yang, Wei Ma, Xidong Pi, and Sean Qian. A deep learning approach to real-time parking occupancy prediction in spatio-termporal networks incorporating multiple spatio-temporal data sources. *arXiv preprint arXiv:1901.06758*, 2019.

Yanxu Zheng, Sutharshan Rajasegarar, and Christopher Leckie. Parking availability prediction for sensor-enabled car parks in smart cities. *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 1–6, 2015.