2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)
October 8-12, 2022, Macau, China

# Reinforcement Learning for Curbside Space Management with Infrastructure Autonomy and Mixed Vehicle Connectivity

Shuyi Yin[1], Zhiyong Cui[2] and Yinhai Wang[1]

2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC) | 978-1-6654-6880-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ITSC55140.2022.9922149

***Abstract*— Urban curbside parking has been a headache for a wide range of urban stakeholders. It is difficult to solve and is rarely regarded independent from the well-studied parking management problem. However, a closer look at its properties and a comparison points out the unique features of curbside parking game that involves both the parking/cruising traffic and the roadway traffic. Two gaps in literature and prototypes that shape the future of the curbside are identified. And to bridge them, this paper proposes to innovatively solve it by infrastructure autonomy, modeling the curbs as agents. Later, this study considers heterogeneity of vehicles in two dimensions and connects them to reduce problem complexity. A model for curbside space management (CSM) is developed and solved via a reinforcement learning (RL) scheme. Partial observations and full information are fed to different components in the model respectively for robust training. Results based on simulation show the proposed model outperform two baseline control strategies and learns robustly.**

## I. INTRODUCTION

The competition for curbside parking spaces has been intense. Authors in [1] concluded that up to 30% of urban traffic during rush hours can be cruising for parking; similarly, [2], [3] found that 34% of congestion in urban areas is due to cruising-for-parking. The emerging demands of e-commerce and shared mobilities only intensify this competition at the curb. In contrast to traditional purposes, i.e., parking for hours, these new sources of demand bring more randomness, such as their spatiotemporal demand patterns and requested duration of occupancy. Curbside parking system is often overwhelmed. As a result, while cruising for parking, the additional vehicle miles traveled (VMT) at low speeds exacerbates congestion and air pollution. Even though cruising traffic is only due to inefficient parking provision, its externalities are endured by the whole traffic system [4].

There has been a good body of literature and proof-of-concepts trying to understand and solve the parking problem, such as traffic-parking interactions [4], [3], parking information management [5], [6], [7], parking demand management [8], [9], and cruising behavior [10]. However, a *key assumption* of theirs is that spaces of parking infrastructure are physically fixed.

In fact, the curbside parking problem can be interpreted as a bi-level problem, where space allocation should come first as the upper level and the well-studied management problem is at the lower level. Therefore, curbside *space management* is a new problem from previously investigated *parking management*. The core task of *space management* is to enable infrastructure with control autonomy for deciding when and how the infrastructure should adjust itself. It is then not difficult to see *two significant gaps* present in state-of-the-art literature and prototypes: *connectivity* and *flexibility*. The former stresses the fact that connectivity is critical for data collection, user confirmation, and routing guidance, e.g., in reservation for curbside parking spaces [11] and in guiding vehicles to allocated spaces [3]. For example, freight and parcel delivery service drivers are typically unaware of nearby parking opportunities during their parking search approaching the destination [12], and it is not surprising to see more than two thirds of all parking choices are illegal in Paris, as pointed out by [13]. The *flexibility gap* emphasizes that infrastructure should be physically flexible to repurpose itself to fit the modern curb demands, i.e., by various types of vehicles with heterogeneous demands.

Connected Vehicle (CV) technology brings hope to bridge the *connectivity* gap, because it facilitates rapid and continuous communication among vehicles and between vehicles (V2V) and transportation infrastructure (V2I) [14]. However, limited by the speed of market adoption, in the near- or mid-future, there will be a mix of CVs and non-CVs in the parking traffic. Hence, assuming ubiquitous connectivity is dangerous in developing theory and models for future curbside system, because in mixed connectivity scenarios, information is never fully retrieved.

Although control models [15] have achieved good results, the non-equilibrium and stochastic nature of curbside parking game undermines their correctness and applicability. Instead, without strong assumptions of arrival patterns, service time, equilibrium or car following behavior, reinforcement learning (RL), formulating the game as a Markov decision process (MDP), is a promising data-driven approach for learning adaptive control policies [16] and for our curbside space control task. It has proven successful in many tasks of the transportation domain, e.g., autonomous driving [17], [18] and traffic signal control (TSC) [19]. Similar to our control problem, TSC problems also looks at how infrastructure can be dynamically adjusted to serve its users. Among RL models, advantage actor-critic (A2C) [20] is a popular group of models as the actor (athlete) and critic (judge) setup allows for feeds of different levels of knowledge of the environment to the two networks and thus the learning is robust [21]. Another thread of TSC research investigates TSC using CV data specifically, but only a few consider mixed connectivity

[1]Shuyi Yin and Yinhai Wang are with the Smart Transportation Application and Research Lab (STAR Lab) at the Department of Civil and Environmental Engineering, University of Washington, 101 More Hall, Seattle, WA 98195. Contact syin1@uw.edu, yinhai@uw.edu

[2]Zhiyong Cui is with the School of Transportation Science and Engineering, Beihang University, China. Contact zhiyongc@buaa.edu.cn

situations [22], [23], [24]. Most notably, [24] discussed why researchers report seemingly conflicting performances of RL-TSC algorithms at low penetration rates, and designed a A2C architecture leveraging CV and full for training actor and critic respectively.

This study investigates the problem of urban curbside space control and answers three challenges: infrastructure autonomy (curbside space control), connectivity, and flexibility. Specifically, we designed an A2C RL model for a curbside space agent that leverages partial CV observation of traffic state, controls the physical allocation of space to CV vs. non-CV, and achieves promising performances over benchmarks. Therefore, the main contributions of this paper include:

- distinguished curbside *space management* from the *parking problem* and solveed *space management* from the angle of infrastructure (spaces), by allowing for infrastructure autonomy;
- presented an A2C algorithm, extended it to allow for multiple vehicle types and mixed connectivity scenarios, and validated the performance of our model at an isolated blockface with synthetic parking demand;

## II. Model

This section first discusses the difference between curbside space management (CSM) problem and traffic singal control (TSC) and how that affects algorithm design. Based on this discussion, a customized RL-CSM model is developed and further improved by allowing mixed connectivity exploiting the algorithm structure.

### A. Preliminaries

The CSM (curbside space management) problem is different from TSC (traffic signal control) problem in three aspects. Firstly, in TSC problems there is only one *flow* – the traffic flow – while in CSM problem, there are two. The parking flow and traffic flow interactive with each other and the failure at the curbside can easily propagate through the roadway network. Secondly, *serving sequence* is different in two problems. In TSC, vehicles are always first-in-first-out (FIFO), and this is why vehicle delay has been widely adopted for environment state [19]; in CSM, as the vehicles are immediately rerouted if rejected for parking, they will not form queues on the roadway (or at least this is not expected), and thus vehicles who arrive late could be served first. Thirdly, *capacity of infrastructure* is different. Approaching lanes' capacities are constant in TSC, and thus it is common to include the lane occupancy rate (number of vehicles divided by lane capacity) in system state or reward, i.e., "pressure" [25]. In contrast, in CSM, the capacity of the infrastructure is dynamic. While vehicles are served by CSM, i.e., parked, they stay in the system before exiting. Hence, the remaining service capacity depends on how much allocated space has been occupied. By serving the vehicles, the service capacities are constantly changing in CSM.

These differences tell that while there are similarities between CSM and TSC and many concepts are transferrable, the specific design of RL-CSM models must be domain-specific.

### B. Blockface Problem and RL-CSM Model

Without loss of generality, CSM problem at an isolated blockface with four curbs is solved in this study, as in Figure 1. The agent is controlling four curbs synchronously and each curb of the blockface is indexed by $i \in I$.

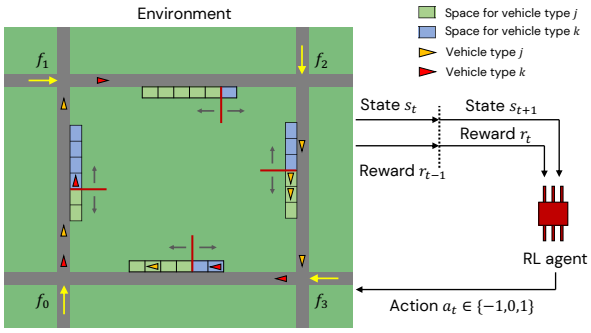



Fig. 1: An agent controls four curbs of the blockface synchronously, accounting for two different vehicle types and interaction with the environment.

**State (Observation)**. The state $s_t$ for the control agent is defined as a vector of (1) space allocated in each curb $i$ for each type of vehicles $j$, i.e., $c_t(i,j)$ for $i \in I, j \in J$; (2) parking occupancy at curb $i$ for vehicle type $j$, i.e., $p_t(i,j)$ for $i \in I, j \in J$; (3) vehicle arrival count at each curb $i$ over the last time step $e_t(i), i \in I,$; and (4) parking failure/reroute $d_t(i), i \in I,$. In summary, the state $s_t$ for this blockface agent has its naive form

$$s_t = [c_t(i,j), p_t(i,j), e_t(i), d_t(i)], \text{ for } i \in I, j \in J, \quad (1)$$

To further reduce the complexity,the curbs are controlled synchronously. Therefore, information can be summarized as follows. Notice number of reroute per arriving vehicle replaces the total reroute count in $d_t$.

$$\begin{aligned} c_t(j) &= \sum_{i \in I} c_t(i,j) \\ p_t(j) &= \sum_{i \in I} p_t(i,j) \\ e_t(j) &= \sum_{i \in I} e_t(i,j) \\ d_t(j) &= \frac{\sum_{i \in I} d_t(i,j)}{\sum_{i \in I} e_t(i,j)} \end{aligned} \quad (2)$$

Consequently, the condensed form of state vector $s_t$ is

$$s_t = \left[c_t(j), p_t(j), e_t(j), d_t(j)\right], \text{ for } j \in J \quad (3)$$

**Action**. In synchronous control, the RL-CSM agent applies the same action to the all curbs: $a_t = a_{t,i}$, for $i \in I$. The action space $A$ is thus {-1,0,1} - either adjust space allocation for vehicle type $j$ by one unit, or do not change the configuration (assuming $|J| = 2$, increasing allocation for one type means shrinking allocation for the other type).

There are some edge cases where action $a$ might not make sense given state $s$: they are mostly due to physical constraints. For example, (1) the agent cannot decrease space allocation for vehicle type $j$ further when all allocated spaces for $j$ are already occupied, and (2) the agent cannot allocate more space to type $j$ when all resources have already been yielded to this vehicle type. These actions are restricted the physical parking constraint that (1) occupancy cannot exceed capacity allocated for every vehicle type $j \in J$ and (2) total space count is constant. In fact, allowing these edge actions might be helpful, as illustrated in Figure 1. Starting from uniform allocation across curbs and executing the same action at each time step, the curbs can reach uneven allocation by taking the edge actions. For example, at curb $i$ space for vehicle type $j$ is all taken, while at another curb $i+1$ there is one unoccupied space for type $j$. The agent can choose $a = -1$ to shrink the allocation for type $j$ and this will not affect curb $i$ because there are not enough space for $j$ to be reallocated. At curb $i + 1$, instead, this reallocation can be applied. This process will allow for more flexibility in controlling the blockface.

**Reward**. The reward $r_t$ is defined as the minus number of reroutes of all vehicles in all curbs. This metric includes information from all vehicle types $j \in J$ because $r_t$ reflects the ground-truth feedback by the environment.

$$r_t = -\sum_{i \in I} d_t(i) \tag{4}$$

, where $d_t(i)$ counts the number of reroutes occurred during time step $t$ at curb $i$. This reward definition echos some RL-TSC studies, where cumulative delay are adopted [19], [24]. In this study, the *delay* of users due to the CSM control is the additional distance they have to cruise. And since the simulated blockface has equal lengths of its four sides, the additional cruising distance is proportional to the number of reroutes.

### C. Algorithm

In this section, the A2C algorithm is introduced and then customized to allow for multiple vehicle types for RL-CSM.

*1) Vanilla A2C:* The Advantage actor-critic (A2C) is characterized by the actor-critic architecture where the critic judges the action performed by the actor [20].

Actor-critic is a "family" of on-policy algorithms where critic, parameterized by $w$, estimates the value function (state value value in our case) and the actor updates its parameter $\theta$ for the policy network following critique from the critic. Deep neural networks are used in both the value network $v(s; w)$ and the policy network $\pi(a|s; \theta)$. The agent observes $o_t$ (could be partial or full of the true environment state $s_t$) at time step $t$ and samples an action $a_t$ based on its policy network $\pi(a|s; \theta)$. Then after executing this action $a_t$, the agent waits and later receives a reward $r_t$ and makes a new observation $o_{t+1}$. Hence, it has collected an experience tuple $(o_t, a_t, r_t, o_{t+1})$. In order to fully utilize past experience, an experience replay buffer will store the experience tuples and will be referred to during training.

The plain policy gradient with baseline is computed as follows, where baseline $b$ can lead to small variance and speed up convergence, if chosen appropriately [16].

$$\frac{\partial V_\pi(s)}{\partial \theta} = \mathbb{E}_{A \sim \pi}\Big[\frac{\partial \ln \pi(A|S; \theta))}{\partial \theta} \cdot (Q_\pi(S, A) - b)\Big] \tag{5}$$

A common choice of baseline $b$ is state value $v(s; w)$, due to its independence of $A_t$ and its closeness with $Q_\pi(S, A)$: $\mathbb{E}[Q_\pi(S, A)] = V_\pi(S_t)$. In addition, the Q-value $Q_\pi(S, A)$ is approximated by sampling experience tuple from the buffer.

$$\begin{aligned} Q_\pi(s_t, a_t) &= \mathbb{E}_{S_{t+1}}[R_t + \gamma \cdot V_\pi(S_{t+1})] \\ &\approx r_t + \gamma \cdot v(s_{t+1}; w) \\ b &= v(s_t; w) \end{aligned} \tag{6}$$

For each episode during training, the actor samples actions according to its policy network $\pi(a|s; \theta)$ until the simulation ends, while adding experience tuples $(o_t, a_t, r_t, o_{t+1})$ for $t \in \mathbb{Z}^+$ to the buffer. After the episode ends, the agent samples experience from the replay buffer and train the networks. First, it will compute target $y_t$ and TD error $\delta_t$:

$$\begin{aligned} y_t &= r_t + \gamma \cdot v(o_{t+1}; w) \\ \delta_t &= v(o_t; w) - y_t \end{aligned} \tag{7}$$

The policy (actor) and value (critic) networks are updated accordingly. And each update performed on the policy network $\pi(a|o; \theta)$ increases the probability of selecting an appropriate action to achieve a Q-value higher than the baseline $v(o_{t+1}; w)$.

$$\begin{aligned} \theta &\leftarrow \theta - \beta \cdot \delta_t \cdot \frac{\partial \ln \pi(a_t|o_t; \theta)}{\partial \theta} \\ w &\leftarrow w - \alpha \cdot \delta_t \cdot \frac{\partial v(o_t; w)}{\partial w} \end{aligned} \tag{8}$$

*2) Extending to RL-CSM customized A2C:* One of the difficulties in modern curbside parking game comes from the various types of vehicles requesting a stop. One dimension is classes of vehicles, e.g., truck, bus, and cars, and the other is connectivity, i.e., CVs vs. non-CVs.

Instead of including both dimensions of complexity, i.e., vehicle class and connectivity, this study seeks to find a connection between them so that parking vehicles in this study have only one identifier. Without loss of generality and assuming uniform physical dimensions, this study puts all curbside parking vehicles into two categories: delivery and general. Delivery vehicles include vehicles that only request a short stop, e.g., buses, taxis, ridesourcing vehicles, and food and goods delivery vehicles, among others. General vehicles may include those that require longer parking duration, e.g., commuters, loading trucks, and emergence vehicles. Furthermore, it is assumed that delivery vehicles are connected while others are not. This characterization comes from the fact that commercial fleets are more likely and faster to adopt new vehicular technologies, such as electrification [26]. Therefore, by categorizing parking vehicles into two general groups, the connection between vehicular connectivity to

vehicle classes is established. And only one dimension of vehicular complexity needs consideration. The extension of the A2C model will be discussed with two vehicle types: CVs and non-CVs.

As the critic evaluates the action taken by the actor and the actor updates its network based on the critic's suggestion, it is natural to question if the actor needs full information from the environment at all to learn robust policy, given the critic has perfect information to guide it. In fact, the actor-critic set-up allows for different levels of information fed to the critic and actor respectively, and this facilitates effective estimation of optimal policy even without full information. In this CSM problem, vehicles of different connectivity (CVs vs non-CVs) naturally point to different levels of data availability to the system. In future mixed-connectivity scenarios, parking non-CVs are hardly detectable in the traffic, because they do not communicate with the infrastructure, and are not identifiable in other sensor data streams (for example, it is nearly impossible to tell if a vehicle is going to park until its destination in videos). Thus, how to map from detectable CV data to full information is challenging. To answer this question, in this CSM problem, the critic is fed with full information, including both CVs and non-CVs info, while the actor is only provided with CVs data. This guidance by critic with perfect knowledge can help the actor learn a robust mapping from partially observable info to the true environment transitions. This could be particularly helpful to stabilize training for low CV penetration scenarios.

The previously discussed state $s_t$ design has two groups of information: that collected at the curbside infrastructure $c_t(j)$ and $p_t(j)$, and that collected from vehicles $e_t$ and $d_t$. Aided by state-of-the-art sensor technologies, the infrastructure can tell if a parked vehicle is CV or not. And thus it is technically feasible to count $p_t(j = \text{CV})$ and $p_t(j = \text{non-CV})$. Additionally, since the system has a record of space allocation, it also knows $c_t(\text{CV})$ and $c_t(\text{CV})$. Therefore, it is $e_t$ and $d_t$ that can contain full information or CV-only data. Due to limited connectivity, there is never a chance of knowing non-CV demand (arrivals). Similarly, when these non-CVs arrive, if all allocated non-CV spaces are occupied, then they are rerouted instantly to somewhere else. It is impossible to know the number of non-CV parking failures (reroutes) neither. This means in future real-world applications, the system will only know CV-data. This justifies previous decision that the actor be only fed with CV data. This does not prevent the simulation from supplying the critic with full information, because after guiding the actor through training, critic will not participate in the execution phase of CSM. State variables $e_t$ and $d_t$ for *critic* can be calculated as

$$\begin{aligned} e_t &= \sum_{i\in I}\sum_{j\in J} e_t(i,j) \\ d_t &= \frac{\sum_{i\in I}\sum_{j\in J} d_t(i,j)}{\sum_{i\in I}\sum_{j\in J} e_t(i,j)} \end{aligned} \tag{9}$$

And the variables for actor are calculated as

$$\begin{aligned} e_t &= \sum_{i\in I}\sum_{j=\text{CV}} e_t(i,j) \\ d_t &= \frac{\sum_{i\in I}\sum_{j=\text{CV}} d_t(i,j)}{\sum_{i\in I}\sum_{j=\text{CV}} e_t(i,j)} \end{aligned} \tag{10}$$

## III. Experiments

This section focuses on curb control problem at a simulated blockface. The set-up of the simulation is described first, followed by a sensitivity analysis on the design parameters, and a comparison is presented last between the proposed RL-CSM algorithm and baseline strategies.

### A. Simulation

**Physical layout**: All sides of the blockface are set of length 100m and with one curb consisting of 10 uniformly sized parking spaces; vehicle sizes are uniform as well. Consequently, every curb space can physically hold up to one vehicle, regardless of its type. The roads are single-laned and the vehicles can only make right turns. Therefore, the vehicles will (1) enter the system at the four corners of the blockface; (2) try parking at the closest curb downstream; (3) make right turns to cruise if parking is rejected; and (4) exit the system at the closest corner only after they have been served. Another parameter that influences the starting state of the system and thus the parking dynamics is the initial space allocation to the two vehicle types: CV vs. non-CV. It is especially important if some fixed strategy is followed by the controller, as the system can soon go into deadlock if not controlled well at the start.

**flow properties**: In order to simulate reasonable demands but to avoid gridlock, turnover rates at a curb are calcualted first. For a 10-space curb, it can theoretically hold $\frac{3600}{3*60}*10 = 200$ vehicles/hr, if requested parking duration is uniformly three minutes per vehicle. This is an certainly overwhelming demand because zero time gaps are assumed between parking events. Uniformly generated at the four corners of the blockface $f_0 = f_1 = f_2 = f_3 = 200$ veh/hr (see Figure 1), total flow rate will be $\sum_{i=1}^{4} f_i = 200 \times 4 = 800$ vehicle/hr for the system. The composition of the flow is determined by the ratio of CVs to non-CVs: a CV-to-non-CV ratio of 1 indicates an equal share. The third parameter is the ratio of background traffic flow rate to parking traffic flow rate. Background traffic makes the simulation more realistic with the traffic dynamics and network effect. The fourth and last parameter is demand arrival pattern. In this study, we adopt uniform demand.

**vehicle properties**: The primary feature of parking demand is parking duration. As this study has linked CVs with delivery vehicles in modern parking game (see section *Extending to RL-CSM customized A2C* for the argument), they will park shorter. On the contrary, non-CVs will park longer. Without loss of generality, parking duration of non-CVs are set to 3 mins, and the ratio of CV parking duration to that of non-CV as a user-specified parameter.

Therefore, the simulation can be established in Simulation of Urban MObility (SUMO) [27] as in Figure 1, and there are

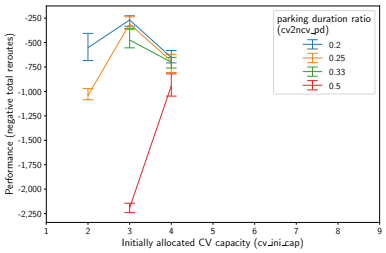
five tunable parameters: **park2curb** (ratio of parking demand to theoretical number of turnovers), **background2park** (ratio of background traffic to parking traffic), **cv2ncv_pf** (ratio of CV count to non-CV count in the parking flow), **cv2ncv_pd** (ratio of CV parking duration to non-CV parking duration), and **cv_ini_cap** (initial allocation per curb to CVs).

### *B. Sensitivity Analysis with Two Baseline Strategies*

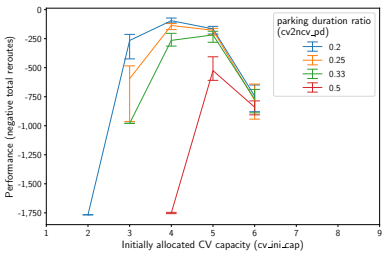
The goal of sensitivity analysis is to explore how baseline models perform in different scenarios and especially when they fail. This study tests a wide range of combinations. The agent's objective is to maximize the negative number of reroutes.

Two simple control strategies serve as baselines in this study: *no-action* and *max pressure*. Following the *no-action* strategy, the curb space controller will not adjust its allocation, and thus service capacity is solely determined by the initial state. In *max pressure* strategy, the agent adjusts based on the "pressure". Analogous to pressure defined in TSC problems [28], [25], pressure in CSM is defined as the difference of failure rate between CVs and non-CVs. Assuming incoming demand similar to the freshly observed, the agent estimates how likely a CV and a non-CV parking is unsuccessful respectively. If the pressure is insignificant, the agent keeps the existing allocation. If the pressure is significant, the agent favors the vehicle type that has seen greater probability of unsuccessful parking request. One thing to notice is max pressure needs full information of the environment, i.e., both CV and non-CV data, to make decisions. This would tremendously limit its applicability in real-world applications as non-CV data and especially their failure rates are practically unquantifiable.

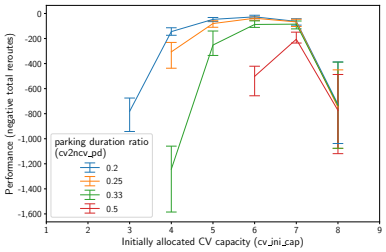
Figure 2 presents system performance following the *no-action* strategy. The curb agent does not adjust its space allocation for CV vs. non-CV whatsoever. Firstly, share of CV flow proves to be important. If there are fewer CVs than non-CVs, i.e., $cv2ncv\_pf < 1$, the agent could only produce deadlocks in the simulation, and this is why Figure 2 only includes results from $cv2ncv\_pf \geq 1$ cases. This tells that the "no-action" strategy cannot answer low connectivity scenarios ($< 50\%$) at all, no matter how resources are provided at the beginning. This could be due to the fact that average parking duration is longer is fewer CVs are preset. Secondly, from all five subplots of Figure 2, it can tell the shorter CVs park ($cv2ncv\_pd$), the fewer reroutes will ensue and thus the better the blockface behaves. The high turnover rate provided by CVs' short stays is also helpful to the system. Thirdly and most importantly, following no-action strategy, by starting with a good allocation, the agent can perform well for the traffic - even though that optimal set of starting space allocation is quite limited. This is demonstrated by the "concave" shape of the connected lines in every subplot of Figure 2.




(a) $cv2ncv\_pf$ = 1




(b) $cv2ncv\_pf$ = 2



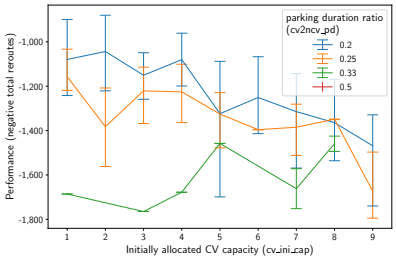

(c) $cv2ncv\_pf$ = 5

Fig. 2: Sensitivity analysis for *no-action* strategy.

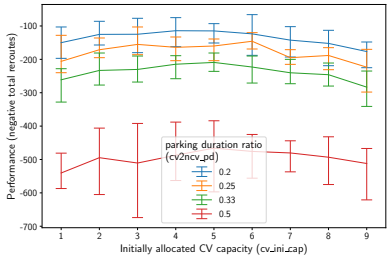
Figure 3 demonstrates how max pressure performs in the simulations. Improvements over the *no-action* strategy is significant in two aspects. First of all, max pressure strategy generates meaningful control results for $cv2ncv\_pf = 0.33$ and $cv2ncv\_pf = 0.5$ scenarios. By adjusting its space allocation, it shows greater ability to account for low connectivity/ long average parking duration scenarios. Second improvement is its balanced and good performances in different scenarios. Rather than the "concave" curve with strong peaks by *no-action* strategy, *max pressure*'s curves are a lot more smoother (see Figure 3). This means regardless of where it started in the space allocation, it can adjust itself to good states. This echos the philosophy of balancing behind minimizing the "pressure" [25]. A third feature in *max pressure*'s performance is the gaps between curves are smaller than the no-action agent. This indicates the max pressure agent are less sensitive to parking duration ratios.

In summary, the max-pressure agent performs (1) better in scenarios where average vehicular parking time is high; (2) more stable even at various and sometimes bad starting points; and (3) less sensitive to parking duration ratios when they are low. This result further justifies why flexibility should be enabled in infrastructure control and makes us

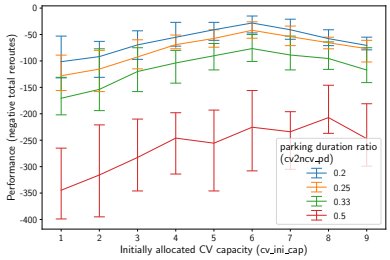
wonder how much the even more flexible RL-CSM agents can outperform these two baselines.




(a) $cv2ncv\_pf$ = 0.33




(b) $cv2ncv\_pf$ = 2




(c) $cv2ncv\_pf$ = 5

Fig. 3: Sensitivity analysis for *max pressure* strategy.

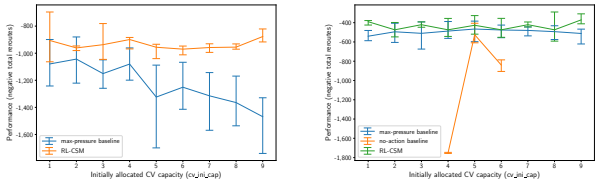
### C. RL Control of the Blockface

The policy network of the RL-CSM agent $\pi(a|s;\theta)$ consists of two hidden layers of $2|o| = 2 \times 6 = 12$ fully connected neurons and the activation function is Rectified Linear Unit (ReLU). Here $|o| = 6$ is the length of the observation vector, as described earlier. The output layer is a Softmax layer with 3 neurons, one for each possible action. The output of the layer is a probability distribution, and the action is sampled accordingly. The value network $v(s;w)$ has the same structure with the policy network, except that the output layer is a linear layer that produces a single value for estimating the value of state. Both networks take input of length $|o|$. The two networks are trained with two separate Adam optimizers with learning rate $1e-3$ that decays exponentially. The replay buffer is sized with $1e4$, and experience tuples are sampled in batches of size 64 for training. Notice gridlock situations are penalized by a reward of $-1e4$ so that agents avoids decisions and states that eventually result in deadlockks.

We present performances in two scenarios, **scenario 1** for low connectivity where both *no-action* and *max-pressure* strategies struggle and **scenario 2** for 67% connectivity but with high vehicular parking duration.

**Scenario 1** is parameterized by $cv2ncv\_pf = 0.33$, $cv2ncv\_pd = 0.2$. Performance of baseline models in Figure 4 echoes that in Figure 2 and Figure 3. *No-action* agent cannot manage the traffic and the system soon becomes congested - there is even no meaningful result to display in Figure 2 or Figure 4(a). Performance of *max-pressure* strategy is acceptable as the blue curve in Figure 3(a) and Figure 4(a). However, its performance is at the best reaching $-1000$. The proposed model RL-CSM, on the other hand, outperforms the max-pressure agent at all possible initial states of space allocation. Moreover, the proposed algorithm learns a smooth result, indicating its flexibility of making adjustments overcoming the impact of initial states and finding the universal optimum.

The similar story holds for **scenario 2**, defined by $cv2ncv\_pf = 2$ and $cv2ncv\_pd = 0.5$. This is a case with good traffic connectivity, but relatively long parking duration. Figure 4(b) shows that on average RL-CSM outperforms *max-pressure* baseline, but the margin is small –the *max-pressure* is already performing satisfacotrily. A closer look at Figure 3(b) defined by the same simulation parameters reveals that *max-pressure*'s performance increases with CV penetration rate. This is explained by reduced parking time as CV takes larger portion in the flow as we increase CV to non-CV ratio. Shorter parking lengths always indicates higher turnover rates and more fluidity in the system. But to notice, the proposed RL-CSM again learns a smooth curve indicating it is not significantly influenced by initial state.




(a) scenario 1 (b) scenario 2

Fig. 4: RL-CSM outperforms baselines in typical scenarios.

Figure 5 presents the learning process of the proposed RL-CSM in a third scenario, defined by $cv2ncv\_pf = 1$ and $cv2ncv\_pd = 0.5$. The proposed RL-CSM was heavily penalized by resulting in deadlock in the first attempt but later gradually learns to avoid it in this scenario characterized by moderate penetration rate of CVs and long parking duration. After 10 epochs of simulation/training, the RL-CSM agents outperforms the two benchmark strategies.

## IV. CONCLUSIONS

In conclusion, this study discussed why *curbside space management (CSM)* problem is different from the generally studied *parking management* problem and why CSM is worth studying. It also built simulation for a synthetic blockface

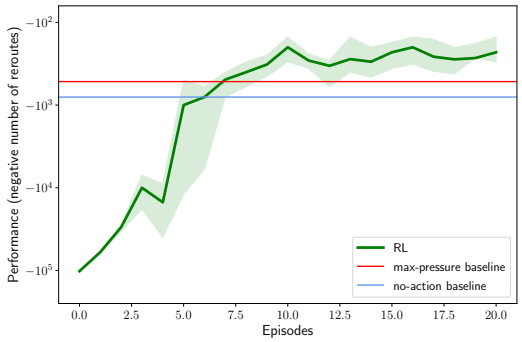



Fig. 5: RL-CSM's learning curve against benchmarks.

CSM problem a customized RL-CSM model where state, action, and reward are all CSM domain-specific. The proposed RL-CSM model extended the general A2C algorithm to a CV-customized model where actor is fed with only CV data and critic knows perfect information. The proposed RL-CSM model with CV connectivity accounted for learned robustly and outperformed the two baseline strategies in the three exhibited scenarios.

This work can be extended in several directions: (1) train and test with more complex demand patterns, and allow unbalanced input at each curb; (2) control the curbs via independent agents instead of one controller for the entire blockface, and solve the derived multi-agent RL problem; (3) use more complex models for policy and value networks, e.g., RNN modules to model historical information for better value approximation and decision making; and (4) include background traffic and more realistic roadway geometries to simulate traffic dynamics.

## REFERENCES


[1] D. C. Shoup, *The high cost of free parking*. Routledge, 2021.

[2] ——, "Cruising for parking," *Transport policy*, vol. 13, no. 6, pp. 479–486, 2006.

[3] S. D. Boyles, S. Tang, and A. Unnikrishnan, "Parking search equilibrium on a network," *Transportation Research Part B: Methodological*, vol. 81, pp. 390–409, 2015.

[4] J. Cao and M. Menendez, "System dynamics of urban traffic based on its parking-related-states," *Transportation Research Part B: Methodological*, vol. 81, pp. 718–736, 2015.

[5] M. I. Idris, Y. Leng, E. Tamil, N. Noor, Z. Razak, *et al.*, "Car park system: A review of smart parking system and its technology," *Information Technology Journal*, vol. 8, no. 2, pp. 101–113, 2009.

[6] Z. Chen, Y. Yin, F. He, and J. L. Lin, "Parking reservation for managing downtown curbside parking," *Transportation Research Record*, vol. 2498, no. 1, pp. 12–18, 2015.

[7] Z. Chen, Z. Xu, M. Zangui, and Y. Yin, "Analysis of advanced management of curbside parking," *Transportation Research Record*, vol. 2567, no. 1, pp. 57–66, 2016.

[8] Z. S. Qian and R. Rajagopal, "Optimal occupancy-driven parking pricing under demand uncertainties and traveler heterogeneity: A stochastic control approach," *Transportation Research Part B: Methodological*, vol. 67, pp. 144–165, 2014.

[9] N. Zheng and N. Geroliminis, "Modeling and optimization of multimodal urban networks with limited parking and dynamic pricing," *Transportation Research Part B: Methodological*, vol. 83, pp. 36–58, 2016.

[10] S. Tang, T. Rambha, R. Hatridge, S. D. Boyles, and A. Unnikrishnan, "Modeling parking search on a network by using stochastic shortest paths with history dependence," *Transportation Research Record*, vol. 2467, no. 1, pp. 73–79, 2014.

[11] X. Wang and X. Wang, "Flexible parking reservation system and pricing: A continuum approximation approach," *Transportation Research Part B: Methodological*, vol. 128, pp. 408–434, 2019.

[12] E. Chaniotakis and A. J. Pel, "Drivers parking location choice under uncertain parking availability and search times: A stated preference experiment," *Transportation Research Part A: Policy and Practice*, vol. 82, pp. 228–239, 2015.

[13] A. Beziat, M. Koning, and F. Toilier, "Marginal congestion costs in the case of multi-class traffic: A macroscopic assessment for the paris region," *Transport Policy*, vol. 60, pp. 87–98, 2017.

[14] X. Di and R. Shi, "A survey on autonomous vehicle control in the era of mixed-autonomy: From physics-based to ai-guided driving policy learning," *CoRR*, vol. abs/2007.05156, 2020. [Online]. Available: https://arxiv.org/abs/2007.05156

[15] C. P. Dowling, L. J. Ratliff, and B. Zhang, "Modeling curbside parking as a network of finite capacity queues," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1011–1022, 2019.

[16] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[17] C. Wu, A. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, "Flow: Architecture and benchmarking for reinforcement learning in traffic control," *arXiv preprint arXiv:1710.05465*, vol. 10, 2017.

[18] E. Vinitsky, A. Kreidieh, L. Le Flem, N. Kheterpal, K. Jang, C. Wu, F. Wu, R. Liaw, E. Liang, and A. M. Bayen, "Benchmarks for reinforcement learning in mixed-autonomy traffic," in *Conference on robot learning*. PMLR, 2018, pp. 399–409.

[19] A. Haydari and Y. Yilmaz, "Deep reinforcement learning for intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[20] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1928–1937.

[21] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1086–1095, 2019.

[22] T. Wu, P. Zhou, K. Liu, Y. Yuan, X. Wang, H. Huang, and D. O. Wu, "Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8243–8256, 2020.

[23] R. Zhang, A. Ishikawa, W. Wang, B. Striner, and O. K. Tonguz, "Using reinforcement learning with partial vehicle detection for intelligent traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, pp. 404–415, 2020.

[24] W. Li, Y. Cai, U. Dinesha, Y. Fu, and X. Di, "Cvlight: Deep reinforcement learning for adaptive traffic signal control with connected vehicles," *arXiv preprint arXiv:2104.10340*, 2021.

[25] H. Wei, C. Chen, G. Zheng, K. Wu, V. Gayah, K. Xu, and Z. Li, "Presslight: Learning max pressure control to coordinate traffic signals in arterial network," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1290–1298.

[26] E. Klock-McCook, S. Li, R. McLane, D. Mullaney, and J. Schroeder, "Ev charging for all: How electrifying ridehailing can spur investment in a more equitable ev charging network," Rocky Mountain Institute, Tech. Rep., 2021.

[27] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Fltterd, R. Hilbrich, L. Lcken, J. Rummel, P. Wagner, and E. Wiessner, "Microscopic traffic simulation using sumo," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2575–2582.

[28] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013.