Reinforcement Learning-based Approach for Vehicle-to-Building Charging with Heterogeneous Agents and Long Term Rewards

Fangqi Liu
Vanderbilt University
Nashville, TN, USA
fangqi.liu@vanderbilt.edu

Rishav Sen
Vanderbilt University
Nashville, TN, USA
rishav.sen@vanderbilt.edu

Jose Paolo Talusan
Vanderbilt University
Nashville, TN, USA
jose.paolo.talusan@vanderbilt.edu

Ava Pettet
Nissan Advanced Technology Center - Silicon Valley
Silicon Valley, CA, USA
ava.pettet@nissan-usa.com

Aaron Kandel
Nissan Advanced Technology Center - Silicon Valley
Silicon Valley, CA, USA
aaron.kandel@nissan-usa.com

Yoshinori Suzue
Nissan Advanced Technology Center - Silicon Valley
Silicon Valley, CA, USA
yoshinori.suzue@nissan-usa.com

Ayan Mukhopadhyay
Vanderbilt University
Nashville, TN, USA
ayan.mukhopadhyay@vanderbilt.edu

Abhishek Dubey
Vanderbilt University
Nashville, TN, USA
abhishek.dubey@vanderbilt.edu

ABSTRACT

Strategic aggregation and utilization of electric vehicle batteries as energy reservoirs to optimize and smoothen power grid demand is highly beneficial to smart and connected communities, particularly large office buildings, which often provide charging access to their employees as part of facilities. Strategic aggregation involves optimizing vehicle charging and discharging to reduce the overall energy bought during peak use (i.e., when energy is more expensive) and the building's net peak demand, which is monitored over an extended period (e.g., a month). This problem entails making sequential decisions under exogenous uncertainty and involves delayed and sparse rewards, a continuous action space, and the complexity of ensuring generalization across diverse conditions. Existing modeling paradigms, e.g., single-shot mixed integer linear programming (MILP), and algorithmic approaches, e.g., heuristic-based strategies, fall short in addressing real-time decision-making under dynamic conditions, and traditional reinforcement learning (RL) models struggle with large state-action spaces, multi-agent settings, and the need for long-term reward optimization. To address these challenges, we introduce a novel RL framework that combines the Deep Deterministic Policy Gradient approach (DDPG) with action masking and efficient MILP-driven policy guidance. Our approach balances the exploration of continuous action spaces to meet user charging demands. Using real-world data from a major electric vehicle manufacturer, we show that our approach comprehensively outperforms many well-established baselines and several scalable heuristic approaches, achieving significant cost savings while meeting all charging requirements. Our results show that the proposed approach is one of the first scalable and general approaches to solving the V2B energy management challenge.

KEYWORDS

Reinforcement Learning, Optimization, Electric Vehicle Charging

ACM Reference Format:
Fangqi Liu, Rishav Sen, Jose Paolo Talusan, Ava Pettet, Aaron Kandel, Yoshinori Suzue, Ayan Mukhopadhyay, and Abhishek Dubey. 2025. Reinforcement Learning-based Approach for Vehicle-to-Building Charging with Heterogeneous Agents and Long Term Rewards . In *Proc.of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 14 pages.

1 INTRODUCTION

The concept of vehicle-to-building (V2B) charging [7, 12] leverages the ability of battery electric vehicles (EVs) to operate as both energy consumers and temporary storage units [24]. V2B systems are particularly relevant in large office buildings, where EVs can be aggregated to optimize energy consumption and reduce peak power demand. By strategically controlling the charging and discharging cycles of EVs, these systems ensure that vehicles meet users' expected state-of-charge (SoC) requirements while minimizing the energy bought during peak time-of-use (ToU) periods [25, 30] and reducing the building's peak power demand over a billing cycle. Implementing this optimization process in practice becomes complex due to the heterogeneity of charging infrastructures [17], the uncertainty of EV arrival and departure times, and the need for a careful balance between energy cost savings and ensuring that the expected final state of charge (SoC) is kept close to user expectation. Additionally, aligning V2B frameworks with complex electricity pricing policies, including both energy and demand charges, adds to the challenge [26, 28]. While prior work has largely modeled this problem as a single-shot mixed-integer linear program [1, 3, 8, 14], such approaches fail to capture the intricacies of real-time decision-making in dynamic environments.

This sequential decision process can be modeled as a Markov Decision Process (MDP); however, solving the MDP presents several difficulties, including delayed and sparse rewards, a continuous

Proc.of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), A. El Fallah Seghrouchni, Y. Vorobeychik, S. Das, A. Nowe (eds.), May 19 – 23, 2025, Detroit, Michigan, USA.

Figure 1: EVs and bidirectional chargers at the research site.

action space, and the need for effective long-term decision-making under uncertainty. To address these challenges, we propose a novel approach to solve this problem that combines the Deep Deterministic Policy Gradient (DDPG) with two key enhancements: action masking and policy guidance through a mixed-integer linear program (MILP). The DDPG algorithm allows us to optimize continuous action spaces while accounting for uncertainties in EV arrival times, SoC requirements, and fluctuating building energy demands. By leveraging action masking, we adjust neural network actions during training using domain-specific knowledge, limiting exploration and guiding the RL agent toward more efficient and feasible policies. The MILP component provides policy guidance during training, steering the RL agent toward near-optimal solutions and enhancing convergence in complex environments. Our approach demonstrates strong generalization across diverse conditions and offers a scalable solution for V2B energy management. Our team includes a large EV manufacturing company that has access to a building with fifteen heterogeneous chargers (Figure 1 shows some of the chargers), and we utilize the real-world charging patterns and building energy data to validate our approach, demonstrating its effectiveness in minimizing monthly energy bills over 9 months (May 2023 - Jan 2024). The summary of our contributions is as follows:

- **Modeling the V2B problem as an MDP with continuous action space**: We model the V2B problem as a Markov Decision Process (MDP) that captures the dynamics of EV SoC levels, varying arrival and departure times, and time-dependent electricity pricing. This formulation addresses delayed and sparse rewards, continuous action spaces, and long-term goals to reduce the monthly peak demand charge and energy costs.
- **Solving the V2B sequential decision-making problem**: We present a reinforcement learning framework based on the Deep Deterministic Policy Gradient (DDPG). We combine DDPG with i) action masking that leverages domain knowledge and the structure of the V2B problem and ii) policy guidance based on solving a deterministic MILP to aid the learning of the optimal policy.
- **Validating with real-world data**: We validate our proposed approach using real-world data from a major electric vehicle manufacturer. The model achieved significant cost savings over nine months (May 2023–January 2024), meeting all user charging demands. Our approach outperforms heuristics and prior work.
- **Ablation Study:** We perform a detailed ablation study to assess the impact of individual techniques, which help us demonstrate the model's effectiveness.

2 PROBLEM FORMULATION

Charger and Time Intervals: Consider the building has N heterogeneous chargers $C = \{C_1, C_2, \dots C_N\}$. Each charger C_i has limits on the charging rate, minimum C_i^{min} and maximum C_i^{max}; $C_i^{min} < 0$ implies the charger C_i is bi-directional and can discharge and $C_i^{min} = 0$ represents a unidirectional charger with no discharging. We assume that all chargers are designed to be able to charge at maximum rates simultaneously, i.e., $\sum_{i=1}^{i=N} C_i^{max} <$ maximum rated capacity of the building. The planning horizon is one billing period, usually a month, which we divide into equal-sized fixed time intervals $\mathcal{T} = \{T_1, T_2, \dots T_{end}\}$, where $T_j - T_{j-1} = \delta$ (we use $\delta = 0.25$ hours). The choice of δ is user-specific and provides a stable decision epoch, preventing rapid changes in the charging rate.

Charging Power: Let us assume that the function $\mathcal{P} : C \times \mathcal{T} \rightarrow \Re$ specifies the power consumed by the charger C_i at time T_j. If the power is zero, the charger is not active, and if the power is negative, the charger discharges, acting as an energy source. Note that by construction $P(C_i, T_j) \in [C_i^{min}, C_i^{max}]$. Let us also assume that function $\mathcal{B} : \mathcal{T} \rightarrow \Re^+$ specifies the average building power consumed in δ time interval. Given the charger and the building power consumption, we can calculate the total cost for the billing period. The parts of the total cost are based on the property type, time of day, and state of the power grid and are based upon the rules and regulations set by the local transmission system operator (TSO) and distribution system operator (DSO). These parts include energy expenses for building power and charging, which vary with peak and off-peak hours, as well as demand charges based on the peak power draw over a longer-term period.

Let the price of the energy consumed is given by $\theta_E : \mathcal{T} \rightarrow \Re^+$ (in \$/kWh). In practice, the Time-of-Use (TOU) electricity rates do not vary continuously and are rather divided into two parts each day, i.e., a peak and a non-peak period. Then, the total cost of the energy consumed is $\Theta_E(\mathcal{P}) = \sum_{j=1}^{j=end} \left(\sum_{i=1}^{i=N} (P(C_i, T_j)) + \mathcal{B}(T_j) \right) \times \theta_E(T_j) \times \delta$. Effectively, Θ_E is a function of charging power $\mathcal{P} = \{P(C_i, T_j) | C_i \in C, T_j \in \mathcal{T}\}$.

Demand Charge: The demand charge is calculated using the maximum (peak) power consumed during any time interval in the billing period, with the demand price denoted as θ_D (in \$/kW). Let $P^{max} = \max_{j=1}^{j=end} (\sum_{i=1}^{i=N} P(C_i, T_j)) + \mathcal{B}(T_j)$ denote the maximum power consumed. The demand charge is given by $\Theta_D(\mathcal{P}) = \theta_D \times P^{max} \times \delta$, which is a function of charging power $\mathcal{P}$. Hence, the total cost of energy bought from the power grid is $\Theta_E(\mathcal{P}) + \Theta_D(\mathcal{P})$. To minimize the cost, we must reduce the net power usage when the cost θ_E is high and manage the power peaks to ensure P^{max} remains as low as possible. Often, the demand charge is levied to ensure that the industrial buildings do not put excess burden on the power grid. In our problem, we use estimates of peak power and denoted it by $\hat{P}^{max}$. It is important to note that the demand charge is typically applied during peak hours of the TOU electricity rate, as reflected in our formulation.

Electric Vehicle Sessions: Assume that during the billing period $\mathcal{T}$, a set of electric vehicles, denoted as $\mathcal{V}$, are serviced at the building. Each EV V is characterized by its arrival time $\mathcal{A} : \mathcal{V} \rightarrow \mathcal{T}$ and departure time $\mathcal{D} : \mathcal{V} \rightarrow \mathcal{T}$. Note that if the same vehicle arrives more than once, we will treat it as a separate session. If the EV arrives between time slots $[T_{i-1}, T_i]$, we consider its effective arrival time as $\mathcal{A}(V) = T_i$. Similarly, if the vehicle departs between $[T_j, T_{j+1}]$, we consider its effective departure time as $\mathcal{D}(V) = T_j$. EV sessions are contiguous, i.e., EV is expected to remain at the site between $\mathcal{A}(V)$ and $\mathcal{D}(V)$, for $\forall V \in \mathcal{V}$. For each V, we know the initial state of charge $SOC^I : \mathcal{V} \rightarrow \Re^+$ and the required final state of charge (measured as a percentage of the battery capacity) $SOC^R : \mathcal{V} \rightarrow \Re^+$ upon arrival. $SOC^{min} : \mathcal{V} \rightarrow \Re^+$ is the minimum allowed SoC for the car i.e., the car cannot be discharged below this value, and $SOC^{max} : \mathcal{V} \rightarrow \Re^+$ is the maximum allowed SoC for the car. The minimum and maximum bounds are specified by the EV manufacturer, considering the impact of charging and discharging on battery health. $CAP : \mathcal{V} \rightarrow \Re^+$ denotes the vehicle's battery capacity in kWh. We track the current SoC of the EV using SOC, where $SOC : \mathcal{V} \times \mathcal{T} \rightarrow \Re^+$ and it is defined later.

Charger Assignment: We define an EV assignment function $\eta : \mathcal{V} \rightarrow C$, where $(V \in \mathcal{V})\ \eta(V) = C_i$ indicates the charger assigned to EV V. Correspondingly, we also maintain a charger-EV occupancy function $\phi : C \times \mathcal{T} \rightarrow \mathcal{V}$, where $\phi(C_i, T_j) = V$, representing the connection of charger C_i with EV V at time T_j. The correlation of these two functions can be expressed as $\phi(\eta(V), T_j) = V,$ s.t. $\mathcal{A}(V) \leq T_j \leq \mathcal{D}(V)$ indicating that if EV V is assigned to charger C_i through the function η, then at any time slot within its stay duration, it is confirmed that EV V is connected to charger C_i. If no EV is connected to the charger at time T_j, the function may return a $\emptyset$ denoting an inactive state, expressed as $\phi(C_i, T_j) = \emptyset$ emphasizing the dynamic nature of the connection function. Note that two vehicles cannot be connected to a charger simultaneously. We consider a first-in, first-out policy that assigns EVs to bidirectional chargers first. A comparison of other approaches is in Table 5 in the appendix. We also maintain that once assigned to a charger, a vehicle is connected to it until departure. For charging the EVs, we approximate a linear charging profile as in prior work [23], and the SoC is updated at time slot T_j using the following equation:

$$SOC(V, T_{j+1}) = SOC(V, T_j) + \frac{P(\eta(V), T_j) \times \delta}{CAP(V)} \tag{1}$$

Feasibility: The set *Feasible* indicates the feasible solutions that satisfy the following constraints:

$$\forall C_i \in C, \forall T_j \in \mathcal{T} : C_i^{min} \leq P(C_i, T_j) \leq C_i^{max} \tag{2}$$

$$\forall C_i \in C, \forall T_j \in \mathcal{T}, \forall V \in \mathcal{V} : SOC(V, T_j) \geq SOC^{min}(V) \tag{3}$$

$$\forall C_i \in C, \forall T_j \in \mathcal{T}, \forall V \in \mathcal{V} : SOC(V, T_j) \leq SOC^{max}(V) \tag{4}$$

$$\forall T_j \in \mathcal{T} : \textstyle\sum_{C_i \in C} P(C_i, T_j) + \mathcal{B}(T_j) \geq 0 \tag{5}$$

Here, Constraint (2) guarantees a valid charging action range, Constraints (3 and 4) ensures that each EV's SoC remains within an acceptable range, and Constraint (5) ensures that discharging power does not exceed building power.

Objectives: One of our objectives for the V2B problem is to minimize the total cost over the billing period, incorporating the Time-Of-Use (TOU) electricity rates and demand charges. This objective is expressed as:

$$\min_{(\eta, \mathcal{P}) \in Feasible} (\Theta_E(\mathcal{P}) + \Theta_D(\mathcal{P})) \tag{6}$$

The second objective ensures that vehicles are charged to their requirement, SOC^R, by the time they leave.

$$\min_{(\eta, \mathcal{P}) \in Feasible} \textstyle\sum_{V \in \mathcal{V}} \max(SOC^R(V) - SOC(V, \mathcal{D}(V)), 0) \tag{7}$$

The inner `max` function ensures EV users' energy requirements are met, even if overcharging occurs. However, in practical scenarios, short stays may make meeting the SoC requirement impossible. To address this, we reformulate the objectives into a multi-weighted framework. The optimal charger assignment and actions are then determined by optimizing these combined objectives.

3 RELATED WORK

We highlight four major challenges of solving the V2B problem, namely: 1) the uncertainty of vehicles and SoC requirements; 2) Time-Of-Use(TOU) pricing, demand charge, and long-term rewards; 3) Heterogeneous chargers and continuous action spaces; and 4) Tracking real-world states and transitions. Below, we briefly cover prior work to tackle these challenges. *A more detailed description of prior work is presented in Table 4 of the appendix.*

Uncertainty of vehicles and SoC requirements. Meta-heuristics and Model Predictive Control (MPC) has been used to solve the EV charging process, focusing on energy cost and user fairness in single-site or vehicle-to-grid (V2G) systems [1, 3, 8, 14]. Studies by Richardson et al. analyze EV charging strategies' impact on grid stability, relevant to V2B systems [20]. Wang et al. proposed a demand response framework for optimizing V2B systems amidst dynamic energy pricing [26]. Additionally, O'Connell et al. utilized Mixed Integer Linear Programming (MILP) to integrate renewable energy sources into grids [16]. However, many of these methods focus on unidirectional chargers and fail to fully account for all exogenous sources of uncertainty (e.g., uncertain arrival and departure times).

Time of use pricing, demand charge, and long-term rewards. Optimizing V2B is hard due to the lengths of billing periods. While prior work (barring some exceptions [8]) optimize and plan for single day horizons [1, 13, 21], they fail to work for longer periods.

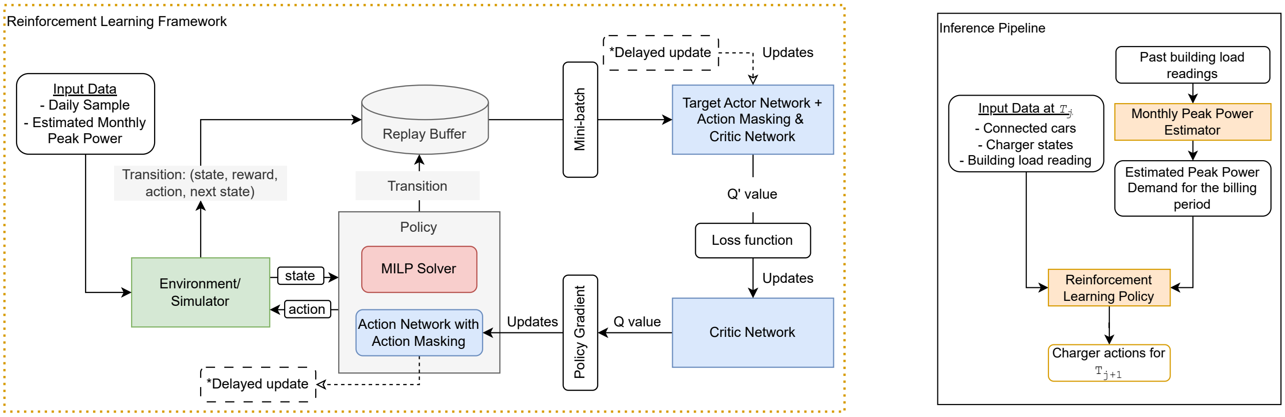
Heterogeneous chargers and continuous action spaces. In practice, buildings develop EV infrastructure gradually, leading to heterogeneous chargers and a more complex action space. While some prior work addresses charger heterogeneity [15, 29], it often neglects long-term rewards (i.e., limit planning to a single day) or fails to account for demand charge, missing the key real-world constraint in the V2B problem. **Tracking real-world state and transition.** Existing solutions validate their approach using simulations with limited interface with the real world (barring some exceptions [8]), thereby making simplistic assumptions that limit deployment.

4 OUR APPROACH

In this section, we discuss the different components in our framework, shown in Figure 2a.

4.1 Markov Decision Process Model

We model the V2B problem as the following MDP.


Reinforcement Learning Framework
Input Data
- Daily Sample
- Estimated Monthly Peak Power
Transition: (state, reward, action, next state)
Replay Buffer
Transition
Mini-batch
*Delayed update
Updates
Target Actor Network + Action Masking & Critic Network
Q' value
Loss function
Updates
Critic Network
Q value
Policy Gradient
Updates
Policy
MILP Solver
Action Network with Action Masking
Environment/ Simulator
state
action
*Delayed update
Inference Pipeline
Past building load readings
Monthly Peak Power Estimator
Estimated Peak Power Demand for the billing period
Input Data at T_j
- Connected cars
- Charger states
- Building load reading
Reinforcement Learning Policy
Charger actions for T_{j+1}

(a) Reinforcement Learning Framework.

(b) Pipeline for Inference.

Figure 2: (a) Our framework relies on daily samples and an estimated monthly peak power. We use RL, i.e., DDPG, and extend it with policy guidance and action masking, to learn a near-optimal policy. (b) At inference time, the model ingests data of connected cars, charger states, building power, and the estimated monthly peak power to make decisions.

State. The complete state space for the problem can be described using features that provide historical, current, and future estimation at a given time T_j, which includes parameters for each vehicle, such as the current SoC, required SoC, departure time, and battery capacity for each EV, along with SoC boundaries across all chargers. Additionally, the current building power, time slot, day of the week, past historical building power, and long-term peak power estimation value are included, resulting in approximately 100 features. We leverage domain-specific knowledge to abstract key information from these features, reducing the state space to the 37 essential state elements.

These features are: **1)** The current time slot, T_j. **2)** The current building power, denoted as $B(T_j)$. **3)** The power gap between the current building power and the estimated peak power for the billing period, given by $\hat{P}^{max}(T_j) - B(T_j)$, where $\hat{P}^{max}(T_j)$ indicates the estimated peak power at T_j, initialized from a value derived from training data. This gap aids the RL model in estimating the optimal peak power for demand charge reduction. **4)** The mean peak building power over the previous 7 days, $\mu(B^H(T_j))$, where $B^H(T_j)$ represents the list of peak building power for the previous 7 days. **5)** The variance of the peak building power over the previous 7 days, $\sigma^2(B^H(T_j))$, helps inform the model about the future building power use. **6)** The day of the week for the current time slot, T_j, which helps the model distinguish daily patterns and enhance generalization. **7)** The number of EV arrivals up to time slot T_j, represented as $|\{V|V \in \mathcal{V}, A(V) \leq T_j\}|$ for tracking EV arrival status. **8)** The energy needed by each EV connected to a charger at time slot T_j, given by $[KWH^R(C_i, T_j)]_{C_i \in C}$, and is initialized to 0. This represents the energy gap between required SoC (SOC^R) and current SoC (SOC) of the EV $V = \phi(C_i, T(j))$, defined as $KWH^R(C_i, T_j) = (SOC^R(V) - SOC(V, T_j)) \times \text{CAP}(V)$. **9)** The remaining time until the departure of each EV connected to the chargers is given by $[\tau^R(C_i, T_j)]_{C_i \in C}$, and is set to 0 when no cars are connected. Each term is computed as $\tau^R(C_i, T_j) = \mathcal{D}(\phi(C_i, T_j)) - T_j$.

Actions. We define the set of actions $\mathcal{A}$, which includes all actions at each time slot T_j with $T_i \in \mathcal{T}$. In this MDP, $\mathcal{A}$ is continuous and specifies the power of all chargers at each time slot T_j, where $A(T_j) = [P(C_i, T_j)]_{C^i \in C}$.

State Transition. States are updated based on actions and EV arrivals/departures at each time slot. To simulate these transitions, we designed an environment simulator that provides and updates states. The state transition function is given as: $Trans(S(T_{j-1})\ A(T_{j-1})) \mapsto S(T_j)$, with the following steps:

(1) Initialize the estimated peak power, $\hat{P}^{max}(T_0)$, which can be derived from historical data (detailed in Section 4) , and update it by $\hat{P}^{max}(T_j) = \max(\hat{P}^{max}(T_{j-1}), \mathcal{B}(T_{j-1}) + \sum_{C^i \in C} P(C_i, T_{j-1}))$, which updates the estimated peak power depending on the previous estimate and the last peak power.
(2) Update SoC of EVs connected to all chargers: $SOC(\phi(C_i, T_j), T_j)$ using action $A(T_{j-1})$ according to Equation (1).
(3) Update the EV charger assignment $\phi(C_i, T_j)$ and $\eta(V)$ by first releasing chargers with departing EVs in the current time slot T_j and then assigning new arrival EVs to idle chargers.
(4) Update the energy requirement of all EVs connected to a charger: $[KWH^R(C_i, T_j)]_{C_i \in C}$ by based on EV's current SoCs.
(5) Update the remaining time of all EVs connected to chargers: $[\tau^R(C_i, T_j)]_{C_i \in C}$ at time slot T_j.

Reward. We define the function $Reward : \mathcal{S} \times \mathcal{A} \longrightarrow \mathfrak{R}$, where $Reward(S(T_j), A(T_j))$ evaluates the reward for actions taken in a specific state, focusing on minimizing the total bill while satisfying SoC requirements. We express reward as $\lambda_S \cdot r_1 + \lambda_E \cdot r_2 + \lambda_D \cdot r_3$ where, $r_1 = \sum_{C^i \in C} \max(0, \min(KWH^R(C_i, T_j), P(C_i, T_j) \times \delta))$, $r_2 = -P(C_i, T_j) \cdot \delta \cdot \theta_E(T_j)$, and $r_3 = -\max(0, \mathcal{B}(T_j) + \sum_{C^i \in C} P(C_i, T_j) - \hat{P}^{max}(T_j)) \cdot \theta_D$. In this reward structure, r_1 promotes actions that charge EVs to reach their required SoC, as intended in Equation (7), while r_2 penalizes the energy cost for the charging actions taken. The third component, r_3, penalizes the increase in demand charges if peak power increases, aligning with our objective in Eq. (6). These functions use three coefficients, λ_S, λ_E, and λ_D to balance trade-offs.

Algorithm 1: Action Masking: $Mask(S(T_j), A(T_j))$.

```
Input: state: $S(T_j)$, action: $A(T_j)$
Output: Masked action: $A'$
1  Initializing: $KWH^R \leftarrow [KWH^R(C_i, T_j)]_{C_i \in \mathrm{C}}$;
     $\tau^R \leftarrow [\tau^R(\phi(C_i, T_j))]_{C_i \in \mathrm{C}}$; $\epsilon \leftarrow 10^{-5}$;
     $C^{max} \leftarrow [C_i^{max}]_{C_i \in \mathrm{C}}$;    $C^{min} \leftarrow [C_i^{min}]_{C_i \in \mathrm{C}}$
   // Mask 1: Set action = 0 if no car is connected
2  $A' \leftarrow \frac{\tau^R}{\tau^R + \epsilon} \times A(T_j)$
   // Mask 2: Stop charging when required SoC is reached for
     uni-directional chargers
3  $A'_{tmp} \leftarrow A'$;    $A'[uniIdx] \leftarrow \min(A'_{tmp}, \frac{KWH^R}{\delta})[uniIdx]$
   // Mask 3: Enforce charging to the req. SoC before departure.
4  $\overline{\mathcal{KW}(T_j)} \leftarrow \frac{KWH^R - (\tau^R - 1) \times C^{max} \times \delta}{\delta}$
     $\overline{\mathcal{KW}(T_j)} \leftarrow \min(\overline{\mathcal{KW}(T_j)}, C^{max})$; $A' \leftarrow \max(A', \overline{\mathcal{KW}(T_j)})$
   // Mask 4: Bidirectional chargers discharge to req. SoC by departure.
5  $\mathcal{KW}^*(T_j) \leftarrow \frac{KWH^R - (\tau^R - 1) \times C^{min} \times \delta}{\delta}$
     $\mathcal{KW}^*(T_j) \leftarrow \max(\mathcal{KW}_t^*, C^{min})$
6  $A'_{tmp} \leftarrow A'$;    $A'[biIdx] \leftarrow \min(A'_{tmp}, \mathcal{KW}_t^*)[biIdx]$
   // Mask 5: Power improvement strategy
7  $powerGap \leftarrow \mathcal{B}(T_j) - \hat{P}^{max}(T_j)$
     $canIncrease \leftarrow ReLU\left(\min\left(\frac{KWH^R}{\delta}, C^{max}\right) - A'\right)$
8  $toImprove \leftarrow \min\left(ReLU(powerGap - \sum A'), \sum canIncrease\right)$
9  $A' \leftarrow A' + \frac{toImprove \times canIncrease}{\sum(canIncrease) + \epsilon}$
   // Mask 6: Do not discharge below building load
10 $toImprove \leftarrow \max(-\mathcal{B}(T_j) - \sum(A'), 0)$
     $negAction \leftarrow ReLU(A' \times -1) \times -1$
11 $A' \leftarrow A' + \frac{toImprove \times negAction}{\sum(negAction) + \epsilon}$
```

4.2 Reinforcement Learning Approach

In this section, we describe the entire reinforcement learning pipeline. We introduce the network structure, discuss how we use a simulator to gather state features and describe the different techniques, such as action masking and policy guidance, used to improve the performance of the V2B problem.

To improve training efficiency, we address the challenge of long state-action sequences by splitting the monthly dataset into daily episodes. This allows the model to capture variations across different weekdays and learn more effectively from shorter episodes, adapting more quickly to daily changes. By incorporating estimated monthly peak power into the state features and reward function, the approach still accounts for monthly demand charges, helping to minimize long-term costs while staying aligned with our objective.

4.2.1 *Enhanced Deep Deterministic Policy Gradient.*

Our approach based on the DDPG framework [11] uses an actor network for continuous actions. During training, we interact with the simulator that provides state abstractions and transitions. To improve RL performance in handling the limitations associated with large continuous action spaces and long-term reward optimization, we introduce action masking and policy guidance techniques. Details of the enhanced approach are in Algorithm 2 in the appendix. Action masking, denoted as $Mask(S(T_j), A(T_j))$, refines the raw actions generated by the actor network by enforcing action validity and utilizing domain-specific knowledge, thereby improving policy performance. Additionally, policy guidance incorporates the MILP solver discussed earlier to provide optimal actions based on current and future information. These optimal actions are stochastically introduced during RL training into the replay buffer (i.e., tossing a biased coin) to mix high-quality actions given a deterministic trajectory with exploratory actions).

4.2.2 *Action Masking.*

Action masking ensures that the policy actions generated by the actor network are feasible during DDPG training. Findings from [4, 6] confirm that differentiable action masking does not interfere with the policy gradient backpropagation process. As a result, the learning process remains effective, while the imposed constraints on the action space prevent the policy from exploring invalid actions, thereby improving training efficiency and optimizing resource usage.

This procedure takes the RL raw action $A(T_j)$, an array of charging power $[P(C_i, T_j)]_{C_i \in C}$ for all chargers, processes it through the following masking steps, and outputs the masked actions A'. Before starting the procedure, we need to obtain the following state features: the remaining power needed to reach the required SoC for all connected EVs (KWH^R), the time remaining for each EV (τ^R), and the maximum ($C^{\max}$) and minimum ($C^{\min}$) power of all chargers (line 1 in Algorithm 1). Also, for our case, since we work with both unidirectional and bidirectional, we denote *uniIdx* and *biIdx* as the indices for unidirectional and bidirectional chargers, respectively. All of the masking techniques referenced below are from Algorithm 1.

- **Mask 1.** We set the charging power $P(C_i, T_j)$ of charger C_i to 0 if no EV is connected, i.e., $\tau^R(\phi(C_i, T_j)) = 0$. (line 2)
- **Mask 2.** Overcharging unidirectional chargers is not beneficial since excess energy cannot be discharged. Thus, we limit the charging power to ensure the SoC of EVs connected to a unidirectional charger remains within their required SoC. For each connected EV, the actions are masked to the minimum of the current charging power and the power needed to reach its required SoC $\left(\frac{KWH^R}{\delta}\right)$ (line 3).
- **Mask 3.** If necessary, we want to adjust actions such that it forces charging to the required SoC before departure to minimize missing SoC, as in Equation (7). We compute the critical power $\overline{\mathcal{KW}^*(T_j)}$, which is the minimum power required for all chargers at time T_j to reach the required SoC of the connected EVs before departing (assuming maximum power C^{max} is utilized in subsequent time slots). The raw action is adjusted if it falls below this value, especially in time slots leading up to the EV's departure (line 4).
- **Mask 4.** This mask is symmetrical to Mask 3 for force discharging. Overcharging bidirectional EVs is only advantageous if excess energy can be discharged during peak hours, but there is no benefit to overcharging just before departure. Using this mask, we force discharge EVs connected to bidirectional chargers, which have excess energy, and they reach the required SoC by departure. Here, $\mathcal{KW}^*(T_j)$ denotes the minimum power to discharge for all chargers $C_i \in C$ at time T_j to guarantee EV can reduce to required SoC when departing (assuming the maximum discharging power C^{min} is utilized subsequently) (lines 5, 6).

- **Mask 5.** We increase charging power while ensuring the masked action stays within the estimated peak power $\hat{P}^{max}(T_j)$. This aims to charge EVs as much as possible towards their required SoC without raising demand charges, thereby avoiding forced charging just before departure, which could elevate peak power. We calculate the "power gap" between estimated peak power and current building power, $\hat{P}^{max}(T_j) - \mathcal{B}(T_j)$. If the current power sum $(\mathcal{B}(T_{j-1}) + \sum_{C^i \in C} P(C_i, T_{j-1}))$ is below this "power gap", we boost the current actions using the available "power" gap, constrained by min $\left(\frac{KWH^R}{\delta}, C^{max}\right)$. (lines 7 to 9).
- **Mask 6.** We adjust the discharging power to prevent cumulatively discharging below the current building power $\mathcal{B}(T_j)$, to satisfy Constraint 5 by reducing the discharging power based on the current actions (lines 10 to 11).

All of the action masking procedures utilize array computations and differentiable operations, such as ReLU [19] and maximum/minimum operations, and the PyTorch framework [18].

4.2.3 *Policy Guidance with MILP Solver.*

Note that for a fixed sample, i.e., a fixed set of EV arrivals and departures, the V2B problem can be modeled as a single-shot mathematical program, i.e., a mixed-integer linear program (MILP), which can solved efficiently (at least, for our problem size) to retrieve the optimal actions. The objective of the MILP is maximizing the multi-objective weighted sum of the total rewards (detailed in Equations 6, (7)), and the other properties of the V2B problem can be encoded as constraints. The fixed sample of arrivals and departures can be extracted from historical data. Naturally, this modeling paradigm does not solve the V2B problem in general—EV arrivals and departures are not known ahead of time—however, this strategy provides a set of optimal actions that the learning module can *learn to imitate*. For our use case, the MILP problem can be solved reasonably fast. For example, for a planning horizon of a day with 15 cars, the problem size averages 800 variables and 1400 constraints and takes 0.05 seconds to solve.

We integrate a MILP solver based on CPLEX [2] to solve the above V2B problem, this solver as a policy guidance sub-routine [10] into the RL training process which feed with current state and the future events to provide optimal next optimal charging action. During training, the MILP solver is stochastically triggered to generate optimal actions based on the current state. These state-action transitions are added to the replay buffer with a predefined coefficient, R^{PG} (see Algorithm 2 in the appendix). The next optimal action is obtained using *MILP*($S(T_j)$, *remainEpisode*), which accounts for remaining events like EV arrivals, SoC requirements, and building power. By blending MILP and RL actor network actions during training, the RL agent explores a more effective action space, improving performance in handling large continuous action spaces and maximizing long-term reward.

4.2.4 *Actor-Critic Network Structure.*

Both the actor and critic networks are fully connected, having two hidden layers with 96 neurons each. Both feature a ReLU activation layer at the end. The critic network outputs a single Q-value estimate, while the actor network outputs the action, which represents the charging power of each charger. To enhance convergence and improve generalization, we normalize all state variables to be within $[0, 1]$ before feeding them into neural networks. Time slot T_j is normalized by division with the number of time slots in a day ($\frac{24}{\delta}$), while power-related variables such as building power $\mathcal{B}(T_j)$, estimated peak power $\hat{P}^{max}(T_j)$ are scaled by their respective statistical values from training data. Furthermore, we normalize the energy capacity $CAP(V)$ of each car by division with the maximum capacity among EVs, $\max(CAP(V))$. For the action $A(T_j) = [P(C_i, T_j)]_{C^i \in C}$, we constrain the output within the range $[-1, 1]$ using the tanh activation function. It is finally translated into the charging power range $[C_i^{min}, C_i^{max}]$ by scaling the value using a constant factor.

4.2.5 *Heuristics and Action Post Processing.*

To enhance the ease of learning in this complex decision space, we use the RL model on weekdays and the peak hours of TOU price within each billing period (for both training and inference). For off-peak hours and weekends, we use a heuristic based on the least laxity task scheduling algorithm (described in Section 5) for off-peak hours and weekends to ensure EVs achieve the required SoC before departure, calculating the minimum charge needed for each time slot. Off-peak hours offer lower electricity prices, allowing for higher EV charging rates, and are excluded from demand charge calculations, making heuristics effective for optimization. Similarly, weekends see fewer EV arrivals and lower power demand, with TSOs excluding them from demand charge assessments. Following the EV manufacturer guidelines, we limit charging to SoC boundaries by clipping the actions of the learned policy within $[SoC^{\min}, SoC^{\max}]$ through post-processing to satisfy Constraints (3) and (4)

4.3 Inference

During execution, our RL-based policy, which is a trained actor network with the action masking procedure, operates at δ time intervals to determine the charging power for all chargers. At each time slot, the state features are generated from data captured from the environment, including charger status (connected EV's current SoC, expected departure time, and SoC), the building's current power and charging rate limits. While we use the estimated peak power $\hat{P}^{max}$ as the state feature based on training samples, as shown in Figure 2b, it can be replaced by any data-driven forecasting or prediction model. Then, we input all the normalized state features, as described in Section 4.1, into the trained RL model to get the charging actions for the next time interval.

5 EXPERIMENTS AND ANALYSIS

To demonstrate the performance of our proposed approach, we use data from our Nissan's research laboratory. We evaluate our approach against several baselines in terms of total bill and peak shaving (demand charge savings).

Data Collection We collected real-world data from Nissan's research laboratory in Santa Clara, California, including building power, EV charger usage, and EV telemetry, over a nine-month period from May 2023 to January 2024. To model the distributions of EV arrivals, SoC requirements, and building power fluctuations, we used Poisson distribution based on historical data. Characteristics of the datasets are shown in Appendix A.2. The number of EVs arriving at the office on weekdays varies daily, illustrating the inherent uncertainties. Arrival and departure hours relative to SoC are depicted in Figure 4 in the appendix, which also presents the distribution of peak power draw and corresponding hours. Main environment parameters are provided in Table 8 in the appendix. We sampled 1000 billing episodes for each month from May 2023 to January 2024.

Downsampling. We varied the number of training samples and found that exceeding a certain limit increased computational demands and worsened performance (see ablation study in Section 5.2). To mitigate this, we applied k-means clustering [5] with $k = 5$ based on the optimal demand charge from the MILP solution, selecting 60 training samples and 50 testing samples per cluster to ensure exclusivity. This approach is further evaluated in the ablation study. The key characteristics of the training and testing data are shown in Table 7 in the appendix.

Estimated Peak Power. To enhance training efficacy, we split the monthly dataset into daily episodes for the model to learn from varying weekday conditions. We include a monthly peak power estimate for each month as an input feature derived from optimal action sequences generated by the MILP solver, using the lower bound of the 99% confidence interval from training data as a conservative demand charge estimate. This input feature is further tuned during RL training.

Hyperparameter Tuning. Hyperparameter tuning is performed on the parameters outlined in Table 6 in the Appendix, which also shows the parameters of the best models selected for each of the nine months. To evaluate the model's performance, we employ a 3-fold cross-validation approach, dividing the 60 monthly training samples into 40 samples for training and 20 samples for evaluation.

Baseline Approaches. To evaluate the performance of our RL approach, we compare it with various methods, including an optimal oracle solution, a real-world charging procedure (baseline approach), and several proposed heuristic approaches. We provide brief descriptions of the baselines here and present the detailed descriptions in Appendix A.3.

- **Optimal MILP Solver (MILP)**: We model deterministic sequences of EV arrivals and departures and solve the problem using the MILP formulation with IBM ILOG CPLEX Optimization Studio [2]. The results serve as an upper bound for comparison, as they utilize an oracle for optimality.
- **Fast Charge (FC)**: This approach simulates current real-world charging procedures, charging all connected EVs as quickly as possible to SOC^{max}.
- **Trickle Charging (Trickle)**: The trickle charging approach utilizes the trickle charging rate, defined as the minimum required charge at each time slot: $P(C_i, T_j) = KWH^R(C_i, T_j)/\tau^R(C_i, T_j)$, to charge all EVs until they reach their required SoC.
- **Trickle Least Laxity First (T-LLF)**: We define the Trickle LLF algorithm (detailed in the Appendix) based on the Least Laxity First approach, a dynamic priority-driven method for scheduling multiprocessor real-time tasks [9]. In EV charging, we define laxity as the difference between the remaining time before departure and the time required to reach the desired SoC at a constant charging rate [27]. For an EV connected to charger C^i at time slot T_j, the laxity value is computed as $(\mathcal{D}(V) - T_j) - KWH^R(C_i, T_j)/C_i^{max}$, with $\phi(C_i, T_j)$ is the EV connected to C_i. At each time slot, we compute the "power gap" (as $\hat{P}^{max}(T_j) - \mathcal{B}(T_j)$), using the estimated peak power and the current building power. This power gap is allocated to all EVs by distributing the trickling charger rate to those prioritized by their laxity.
- **Trickle Early Deadline First (T-EDF)**: We propose the Trickle EDF algorithm in a similar manner to Trickle LLF, with the only difference being the prioritization method. Trickle EDF follows the Early Deadline First approach (based on time of departure of an EV), which was originally designed as a dynamic scheduling algorithm for real-time systems [22].
- **Charge First Least Laxity First (CF-LLF)**: We compute the available "power gap", as in Trickle LLF. Then we calculate the sum of the trickle charging rates for all EVs at the current time slot; if this sum is less than the available "power gap", we have capacity for overcharging. We first assign the charging rate for all EVs to be their trickle charging rates, and then, we charge EVs connected to bi-directional chargers to reach their maximum SoC, following the reverse order of their laxity until the power gap is consumed. If the trickle sum exceeds the power gap, bidirectional EVs are discharged, also based on reverse laxity, to fill the negative gap before resuming the trickle charging. See Algorithm 4 in the appendix.
- **Charge First Deadline First (CF-EDF)**: This follows the same procedure as Charge First LLF but utilizes a different prioritization metric, focusing on the remaining time before EV departure.

5.1 Results

We evaluate all approaches using two metrics: 1) **Peak Shaving:** It is the difference in demand charge between (i) the building's power usage (without any charging) and (ii) by adding charging the EVs under the respective policies. Positive values indicate that the policy reduced the demand charge by controlling the charging actions. 2) **Total Bill:** The sum of electricity cost and demand charge over the billing period, computed by Eq. (6). Note that **missing SoC**—the energy shortfall between required and actual SoC at departure for each EV—is a key metric in the V2B problem. Our RL model, with action masking, applies force charging and discharging in **Mask 2** and **Mask 3** to ensure all EVs reach the required SoC before departure, minimizing missing SoC. For fairness, we apply these force procedures across all proposed heuristics, so we will not report this metric separately, as missing SoC is minimized for all EVs.

We assess the RL model's long-term performance from May 2023 to January 2024, comparing it against baseline approaches on 50 testing samples. Table 1 compares the total bill over nine months across different policies. While MILP offers an oracle-based optimal solution, it is impractical for real-world use and serves as a performance upper bound. The results show that the trained RL model consistently achieves the lowest total bills from May 2023 to January 2024 (except June 2023), outperforming other real-time policies in eight of the nine months and significantly reducing costs compared to the real-world Fast Charge procedure as detailed in Table 1. Table 2 shows the peak shaving achieved by different policies. Our RL approach achieves the highest peak shaving in eight months, barring June. Additionally, heuristic approaches using the First Charge logic, like First Charge LLF or EDF, consistently result in relatively lower total bills and demand charges compared to other heuristics. This indicates that the First Charge approach is effective

Table 1: Total bill of test set, best values are bold. MILP shows the optimal solution given an oracle input. Lower is better.

Policy	MAY	JUN	JULY	AUG	SEP	OCT	NOV	DEC	JAN
MILP	6201.1±50	6713.3±61	7371.0±40	9308.9±51	7231.0±36	7640.6±66	6625.9±42	6079.8±54	6495.1±55
RL	**6222.6±26**	6857.1±122	**7392.2±51**	**9363.3±81**	**7243.0±24**	**7696.3±71**	**6654.9±61**	**6243.7±158**	**6635.0±80**
CF-LLF	6245.9±32	**6843.4±42**	7396.8±26	9435.8±47	7284.1±41	7742.1±48	6675.9±32	6261.8±99	6646.3±81
CF-EDF	6247.6±34	6849.6±48	7399.0±28	9436.1±47	7289.5±48	7747.6±49	6676.3±31	6276.6±87	6639.9±69
T-LLF	6310.7±66	6920.0±75	7432.6±34	9537.5±52	7326.9±48	7800.1±48	6796.9±46	6344.5±132	6670.3±79
T-EDF	6326.6±58	6920.0±56	7455.4±34	9543.0±54	7364.5±48	7819.7±57	6809.7±42	6356.4±88	6673.2±60
Trickle	6333.8±44	6955.6±46	7506.0±37	9570.8±53	7402.1±47	7844.1±60	6842.9±44	6393.1±60	6706.8±53
FC	6308.7±50	6968.6±72	7537.3±83	9541.7±61	7403.6±81	7804.0±69	6813.0±70	6646.9±144	6706.4±77

Table 2: Peak shaving on test set, best values in bold. MILP shows the optimal solution given an oracle input. Higher is better.

Policy	MAY	JUN	JULY	AUG	SEP	OCT	NOV	DEC	JAN
MILP	52.4±46	147.0±59	19.1±32	194.6±58	24.8±31	94.5±67	120.5±52	103.9±49	77.9±59
RL	**31.7±28**	3.8±61	**-1.7±46**	**140.3±97**	**13.1±14**	**39.8±72**	**92.6±67**	**-59.2±152**	**-61.2±77**
CF-LLF	8.3±34	**17.4±43**	-6.3±21	67.9±41	-28.0±28	-6.5±38	71.1±37	-77.2±92	-72.4±83
CF-EDF	6.5±33	11.2±50	-8.5±24	67.5±41	-33.4±34	-12.0±40	70.7±38	-91.9±81	-65.9±68
T-LLF	-56.7±62	-59.2±73	-42.1±40	-33.9±40	-70.9±42	-64.6±41	-50.1±38	-160.2±126	-96.6±76
T-EDF	-72.6±54	-59.3±53	-64.9±30	-39.4±41	-108.5±38	-84.2±52	-62.8±37	-172.1±82	-99.5±58
Trickle	-79.8±39	-94.9±46	-115.5±36	-67.1±34	-146.1±38	-108.6±48	-96.1±36	-208.8±53	-133.2±49
FC	-57.1±46	-113.7±70	-151.3±85	-40.8±42	-155.1±73	-76.0±59	-70.8±72	-473.5±138	-140.3±76

Table 3: Ablation results for final total bill on three months. Lower is better.

RL	**RL\500**	**RL\C**	**RL\F**	**RL\E**	**RL\P**	**RL\A**	**Random\A**
20471.9±137	20494.8±174	20511.6±184	20594.1±181	21130.2±214	21157.0±204	21273.7±209	21627.3±180

in balancing the charging and discharging process, offering better overall performance across all heuristics.

5.2 Ablation Study

We evaluate the contributions of key techniques in our approach through ablation. For the ablation studies, we trained RL models on monthly samples of three months, May to July 2023, and tested their performance on the total bill. The ablations explored are: 1) **RL\500** , RL training with more (500) training samples. 2) **RL\C** , RL training using 60 randomly selected samples from 1000 generated samples. 3) **RL\F** , RL models trained using the complete set of 100 state features defined in Section 4.1. 4) **RL\E**, RL training where the monthly estimated peak power is set to 0, removing the influence of long-term peak power estimation. 5) **RL\P**, RL training without policy guidance. 6) **RL\A**, RL training without action masking, except for forced charging and discharging (Masks 2 and 3), which are retained to minimize missed SoC. 7) **Random\A** , where actions are randomly selected instead of using a trained actor network, followed by action masking. We present the sum of the monthly total bills from May to July 2023 for all approaches in the ablation study in Table 3 and Appendix A.3. Next, we discuss the significance of each ablated feature.

Downsampling. We evaluate the impact of downsampling using k-means clustering to generate 60 training samples from a pool of 1000. The **RL\500** approach, which uses 500 samples, showed no improvement in performance but increased computational burden during training. We also tested **RL\C** , where samples were randomly selected instead of clustered, resulting in a performance drop. These findings confirm that our downsampling method maintains RL performance while improving efficiency.

State abstraction. We then examine the **RL\F** approach, which performs worse, suggesting that condensing state features with domain-specific knowledge improves training and leads to better outcomes.

Policy guidance. The **RL\P** approach, which removes policy guidance, results in decreased performance, highlighting its importance in optimizing actions during training. This guidance narrows down the action exploration space, directing the model toward better solutions.

Long-term prediction. The **RL\E** approach shows worse results, highlighting the importance of accurate long-term peak power estimation during training. This value is used in action masking to improve the charging actions without increasing the monthly peak power and influences the reward function by penalizing actions that raise peak power. When set to 0, the RL model fails to converge to a good global optimum, emphasizing the critical role of peak power estimation in achieving optimal performance.

Action Masking. Training without action masking procedure in **RL\A** leads to a significant performance drop, demonstrating its importance in improving RL performance. This also highlights the challenge of training RL models with 15 chargers in a continuous action space. Action masking incorporates heuristics to guide actions, resulting in significant improvements.

RL actor network. To assess the impact of the actor network, we replaced it with a random policy in the **Random\A** approach. It generates random charging actions before passing through the action masking. The poor performance shows that action masking alone is insufficient, emphasizing the critical role of the actor network in achieving optimal outcomes. Despite all proposed heuristics (except Fast Charge) adhering to action masking constraints, including forced charging and power improvements, the RL approach consistently delivers better results, underscoring the importance of the actor network.

6 CONCLUSION

We propose an RL-based approach to address V2B challenges in smart buildings by optimizing charging power for mixed-mode EV chargers. The goal is to minimize overall costs, including energy bills and demand charges, while ensuring EVs reach their required SoC. Our solution addresses key challenges such as multi-agent decision-making, centralized control of up to 15 chargers, and continuous charging power adjustments, all aimed at minimizing the total energy bill over a month. We evaluate our approach against several heuristic algorithms in simulated V2B scenarios using real-world data from an EV manufacturer. Results show that our trained models effectively manage online EV charging, lowering monthly total bills while meeting SoC requirements.

7 ACKNOWLEDGEMENT

This material is based upon work sponsored by the National Science Foundation (NSF) under Award Numbers 1952011 and 2238815 and by Nissan Advanced Technology Center-Silicon Valley. Results presented in this paper were obtained using the Chameleon Testbed supported by the NSF. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF or Nissan.

REFERENCES

[1] Omid Ardakanian, Catherine Rosenberg, and S. Keshav. 2013. Distributed control of electric vehicle charging. In *Proceedings of the Fourth International Conference on Future Energy Systems* (Berkeley, California, USA) *(e-Energy '13)*. Association for Computing Machinery, New York, NY, USA, 101–112. https://doi.org/10.1145/2487166.2487178

[2] IBM ILOG Cplex. 2009. V12. 1: User's Manual for CPLEX. *International Business Machines Corporation* 46, 53 (2009), 157.

[3] Sara Deilami, Amir S. Masoum, Paul S. Moses, and Mohammad A. S. Masoum. 2011. Real-Time Coordination of Plug-In Electric Vehicle Charging in Smart Grids to Minimize Power Losses and Improve Voltage Profile. *IEEE Transactions on Smart Grid* 2, 3 (2011), 456–467. https://doi.org/10.1109/TSG.2011.2159816

[4] Shengyi Huang and Santiago Ontañón. 2020. A closer look at invalid action masking in policy gradient algorithms. *arXiv preprint arXiv:2006.14171* (2020).

[5] Abiodun M Ikotun, Absalom E Ezugwu, Laith Abualigah, Belal Abuhaija, and Jia Heming. 2023. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences* 622 (2023), 178–210.

[6] Anssi Kanervisto, Christian Scheller, and Ville Hautamäki. 2020. Action space shaping in deep reinforcement learning. In *2020 IEEE conference on games (CoG)*. IEEE, 479–486.

[7] Willett Kempton and Jasna Tomić. 2005. Vehicle-to-grid power fundamentals: Calculating capacity and net revenue. *Journal of Power Sources* 144, 1 (2005), 268–279.

[8] Zachary J. Lee, George Lee, Ted Lee, Cheng Jin, Rand Lee, Zhi Low, Daniel Chang, Christine Ortega, and Steven H. Low. 2021. Adaptive Charging Networks: A Framework for Smart Electric Vehicle Charging. *IEEE Transactions on Smart Grid* 12, 5 (2021), 4339–4350. https://doi.org/10.1109/TSG.2021.3074437

[9] Joseph Y T Leung. 1989. A new algorithm for scheduling periodic, real-time tasks. *Algorithmica* 4 (1989), 209–219.

[10] Sergey Levine and Vladlen Koltun. 2013. Guided Policy Search. In *Proceedings of the 30th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 28)*, Sanjoy Dasgupta and David McAllester (Eds.). PMLR, Atlanta, Georgia, USA, 1–9. https://proceedings.mlr.press/v28/levine13.html

[11] Timothy P Lillicrap et al. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[12] Henrik Lund and Willett Kempton. 2008. Integration of renewable energy into the transport and electricity sectors through V2G. *Energy Policy* 36, 9 (2008), 3578–3587.

[13] Elena Mocanu, Decebal Constantin Mocanu, Phuong H. Nguyen, Antonio Liotta, Michael E. Webber, Madeleine Gibescu, and J. G. Slootweg. 2019. On-Line Building Energy Optimization Using Deep Reinforcement Learning. *IEEE Transactions on Smart Grid* 10, 4 (2019), 3698–3708. https://doi.org/10.1109/TSG.2018.2834219

[14] Joy Chandra Mukherjee and Arobinda Gupta. 2015. A Review of Charge Scheduling of Electric Vehicles in Smart Grid. *IEEE Systems Journal* 9, 4 (2015), 1541–1553. https://doi.org/10.1109/JSYST.2014.2356559

[15] Ajay Narayanan, Srinarayana Nagarathinam, Prasant Misra, and Arunchandar Vasan. 2024. Multi-agent Reinforcement Learning for Joint Control of EV-HVAC System with Vehicle-to-Building Supply. In *Proceedings of the 7th Joint International Conference on Data Science & Management of Data (11th ACM IKDD CODS and 29th COMAD)* (Bangalore, India) *(CODS-COMAD '24)*. Association for Computing Machinery, New York, NY, USA, 332–341. https://doi.org/10.1145/3632410.3632421

[16] Niamh O'Connell, Qiuwei Wu, Jacob Østergaard, Arne Hejde Nielsen, Seung-Tae Cha, and Yi Ding. 2010. Integration of renewable energy sources using microgrids, virtual power plants and the energy hub approach. *IEEE Power and Energy Magazine* 8, 6 (2010), 37–44.

[17] Hyunwoo Park and Chungmok Lee. 2024. An exact algorithm for maximum electric vehicle flow coverage problem with heterogeneous chargers, nonlinear charging time and route deviations. *European Journal of Operational Research* 315, 3 (2024), 926–951.

[18] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[19] Andrinandrasana David Rasamoelina, Fouzia Adjailia, and Peter Sinčák. 2020. A review of activation function for artificial neural network. In *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, 281–286.

[20] Peter Richardson, Damian Flynn, and Andrew Keane. 2011. Electric vehicle charging and grid constraints: A review. *IEEE Transactions on Power Systems* 27, 1 (2011), 544–551.

[21] Nasrin Sadeghianpourhamami, Johannes Deleu, and Chris Develder. 2020. Definition and Evaluation of Model-Free Coordination of Electrical Vehicle Charging With Reinforcement Learning. *IEEE Transactions on Smart Grid* 11, 1 (2020), 203–214. https://doi.org/10.1109/TSG.2019.2920320

[22] John A. Stankovic, Krithi Ramamritham, and Marco Spuri. 1998. *Deadline Scheduling for Real-Time Systems: Edf and Related Algorithms*. Kluwer Academic Publishers, USA.

[23] Olle Sundström and Carl Binding. 2010. Optimization methods to plan the charging of electric vehicle fleets. In *Proceedings of the international conference on control, communication and power engineering*. Citeseer, 28–29.

[24] Jasna Tomic and Willett Kempton. 2007. Using fleets of electric-drive vehicles for grid support. *Journal of Power Sources* 168, 2 (2007), 459–468.

[25] Charles G. Tse, Ben Maples, and Stephen Frank. 2014. The Use of Plug-In Hybrid Electric Vehicles for Peak Shaving. *ASME Digital Collection* (2014). https://asmedigitalcollection.asme.org/energyresources/article/138/1/011201/442565/The-Use-of-Plug-In-Hybrid-Electric-Vehicles-for

[26] Zeyu Wang, Babak Asghari, and Ratnesh Sharma. 2017. Stochastic demand charge management for commercial and industrial buildings. In *2017 IEEE Power and Energy Society General Meeting*. 1–5. https://doi.org/10.1109/PESGM.2017.8274175

[27] Yunjian Xu, Feng Pan, and Lang Tong. 2016. Dynamic scheduling for charging electric vehicles: A priority rule. *IEEE Trans. Automat. Control* 61, 12 (2016), 4094–4099.

[28] Yuna Zhang and Godfried Augenbroe. 2018. Optimal demand charge reduction for commercial buildings through a combination of efficiency and flexibility measures. *Applied Energy* 221 (2018), 180–194.

[29] Zixuan Zhang, Yuning Jiang, Yuanming Shi, Ye Shi, and Wei Chen. 2022. Federated Reinforcement Learning for Real-Time Electric Vehicle Charging and Discharging Control. In *2022 IEEE Globecom Workshops (GC Wkshps)*. 1717–1722. https://doi.org/10.1109/GCWkshps56602.2022.10008598

[30] Song Zhao et al. 2023. Research on peak-shaving of electric vehicle auxiliary power grid considering electric vehicle charging and discharging parameters. In *Proceedings of the Second International Conference on Energy, Power, and Electrical Technology (ICEPET 2023)*. SPIE. https://doi.org/10.1117/12.3004399

A APPENDIX

A.1 Related Work

We provide a more detailed review of prior work here. Table 4 describes the key papers and summarizes their gaps.

Uncertainty of vehicles and SoC requirements. Prior work has taken Mukherjee and Gupta defines this taking into consideration different mobility aspects such as the arrival/departure time of an EV at/from a charging station, trip history of EVs, and unplanned departure of EVs [14]. Empirical studies, such as those by Richardson et al., have analyzed EV charging strategies and their impact on grid stability, which are closely related to V2B systems [20]. The challenges of optimizing V2B systems, especially given the dynamic nature of energy pricing and vehicle usage patterns, have also been addressed by Wang et al., who proposed a demand response framework for smart grids [26]. Additionally, O'Connell et al. applied optimization algorithms, such as Mixed Integer Linear Programming (MILP), to integrate renewable energy sources into grid systems [16]. Other approaches, including meta-heuristics and Model Predictive Control (MPC), have been explored to optimize the smart EV charging process for electric vehicles (EVs), focusing on energy cost and user fairness in single-site or vehicle-to-grid (V2G) systems [1, 3, 8, 14]. However, many of these methods focus on unidirectional chargers and fail to fully account for uncertainty including, vehicle arrivals and departures [14].

Time of use pricing, demand charge, and long-term rewards. Optimizing V2B is a complex problem, made more complex when the lengths of billing periods set by TSOs are considered. Several approaches [1, 13, 21] only optimize and plan for single-day horizons. Lee et al. are able to achieve one-month planning horizons while considering demand charge [8]. However, they assume a homogeneous set of chargers. Thus preventing them from fully realizing the effect of EVs on potential savings.

Heterogeneous chargers and continuous action spaces. Approaches that solve EV charging without considering the ability of EVs to discharge ignore even more potential savings. However, addressing this introduces further complexity to the system. Narayanan et al. works around the limitations of charger homogeneity by using Deep RL [15]. They consider SoC requirements and address the mobility-aware needs of EVs. However, their approach does not consider long-term rewards, instead limiting their planning to a single day. Improving upon these initial approaches, Zhang et al. investigated federated RL for EV charger control, aiming to maximize user benefits [29], and minimize electricity prices. Their approach explores the continuous action space of charging power and extends their planning horizon to an entire week. While their approach includes both discharging and charging actions, they fail to capture the idea of demand charge into their problem, which is critical in the industrial context.

Tracking real-world state and transition. Existing approaches validate their approaches using simulations that have limited interface with the real world. Lee et al. utilizes an existing Adaptive Charging Network (ACN) EV charging testbed along with a mobile application to capture EV telemetry and charger behavior [8]. Thus, they capture the complexity of real charging systems, including battery charging behaviors.

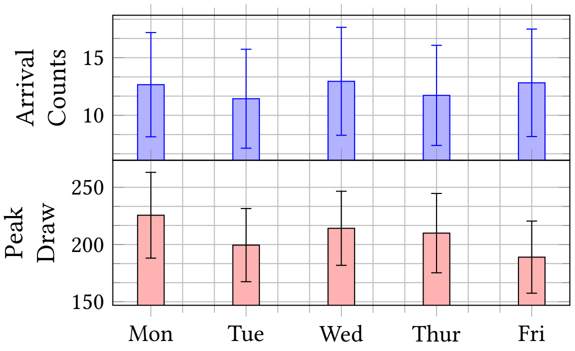


Figure 3: (Top) Arrival counts per day of the week across 8 months. Most of the cars arrive during Wednesday and Friday. (Bottom) Peak building draw per day of the week across 8 months. Mondays and Fridays typically exhibit the highest and lowest power draws.

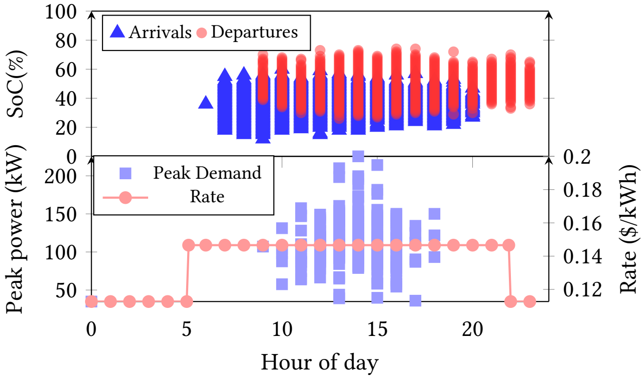


Figure 4: (Top) Distribution of EV hours and SoC for both arrivals and departures across 8 months. (Bottom) Distribution of peak building power draw against the hours of day it was drawn and the TOU rates across the day. The majority of the car arrivals and departure occur within high peak pricing times and all of the peak power draw happens during peak pricing time.

A.2 Complementary Figures

The characteristics of the training set are shown in Figure 3 and Figure 4. Figure 3 show the variations in arrival counts (top) and peak building draw (bottom) across the different days of the week across 8 months. Most cars arrive on Wednesdays and Fridays while the peak building draw is the least on Fridays. Figure 4 (top) show distributions of car arrivals and departures hours against the EVs arrival SoC and required SoC upon departure. Figure 4 (bottom) show the distribution of peak power draw and the hour of day. The line in red signifies the TOU rates across the day. All of the cars arrive within peak hours, while the majority of them leave within the peak hours. Finally, all off-the-peak power draws occur at peak hours. The intersection of these arrivals, departures, and peak power draw hours represents the potential space that our policy can operate on.

Table 4: Comparison of state-of-the-art approaches for EV charging problem with our approach.

Reference	Approach	Objective	Action Space	Planning Horizon	Discharge	Mobility	Req. SoC	Demand Charge
Ardakanian et al. [1]	Distributed control algorithm	EV charging Fairness Allocation	Continuous power rate of 2 chargers	Single day				
Deilami et al. [3]	Rule-based control	Minimize energy cost and grid energy losses	Continuous power rate	Permanent				
Lee et al. [8]	Scheduling algorithm	Minimize demand charge, total load variation, and capacity distribution fairness	Continuous power rate of 80 chargers	One month		✓		✓
Mocanu et al. [13]	Deep Q-learning, Deep Policy Gradient	Minimize building energy cost	Boolean decision for turn on/off 3 devices	Single day				✓
Sadeghianpourhamami et al. [21]	RL: off-policy value-iteration	Minimize power consumption and unfinished charging	Boolean decision (charge or not) on 50 charger stations	Single day		✓	✓	
Narayanan et al. [15]	Deep RL: PPO	Minimize energy bill and satisfy user QoS	Continuous power rate of an EV and a HVAC	Single day	✓	✓	✓	
Zhang et al. [29]	Federated RL: Soft Actor and Critic	Maximize EV user benefits and electricity prices	Continuous power rate of 3 chargers	One week	✓	✓	✓	
Our Approach	DDPG with action masking and policy guidance	Minimize demand charge, electricity cost and missing SoC	Continuous power rate of 15 chargers	One month	✓	✓	✓	✓

Environment Simulator. We process these training data into input samples for our digital twin/environment simulator. We model a digital twin for the target environment and provide several interfaces that allow both simulated and real-world components to leverage our proposed approach. This allows us to investigate how any action or decision can potentially impact the real world. There are two main decisions that must be taken when solving the V2B charging problem. (1) charger assignments and (2) charger actions.

Table 5: Charger Assignment and tiebreaker comparisons with an MILP policy. Assigning to Bidirectional chargers first and then breaking ties by assigining them to the EV that departs later results in the lowest bill. Lower is better.

Assignment	Tie Breaker	Average Monthly Bill ($)
Bidirectional	**Departure**	7037.178±957.869
Bidirectional	Capacity	7037.180±958.867
Bidirectional	Random	7037.647±958.656
Random	Random	7038.770±971.310
Random	Departure	7039.723±968.603
Random	Capacity	7040.216±966.708
Unidirectional	Random	7122.884±981.666
Unidirectional	Departure	7123.066±981.620
Unidirectional	Capacity	7123.073±981.612

Charger assignment. We consider a first-in, first-out policy that assigns EVs to bidirectional chargers first, breaking ties assigning to later departing cars; a comparison of different policy combinations is shown in Table 5. We observe that bidirectional charging assignments outperform all other policies. Tie-breaking strategies that prioritize later-departing vehicles show a marginal advantage. While these assignment policies could be further optimized, we chose to follow this heuristic and focus on the second decision problem: determining charger actions. **Charger actions.** We provide several policies with our simulator to compare our proposed approach. Charger action policies receive a state of the environment for a particular time and generate actions based on this.

A.3 Additional Details on the Approach

Our proposed approach, as outlined in Algorithm 2, is based on the DDPG algorithm [11], which utilizes an actor network to generate actions. Tuples of state, action, reward, and next state are stored as transitions in the replay buffer (lines 9 to 12). During training, we interact with the environment simulator. We provide input from the training dataset. Table 7 shows the characteristics of this dataset. The environment simulator abstracts state features for the RL models and manages state transitions based on the function described in Section 4.1. In each training iteration, we batch state transitions from the replay buffer for model training (line 13). Specifically, DDPG maintains target networks for both the actor and critic, which are used to generate the next state and compute Q-values essential for calculating the critic loss during training. The critic network is trained using gradient descent by minimizing the mean squared error between predicted Q-values and target Q-values derived from the Bellman equation (lines 14-16). The critic learns Q-values for state-action pairs, which are then used to train the actor network through a policy gradient approach (lines 17 and 18). The updates to the target networks are delayed to stabilize the training process (lines 18-19).

To improve RL performance in handling the limitations associated with large continuous action spaces and long-term reward optimization, we introduce action masking and policy guidance

```
Algorithm 2: Improved DDPG with Action Masking and Policy Guidance.
    Input: Initial policy parameters for actor network $\zeta_a$, critic
           parameters $\zeta_c$, target network parameters $\zeta'_a, \zeta'_c$
    Training parameters: actionNoise, $R^{PG}$, bufferSize, batchSize,
    maximum iterations: $M$, training steps: trainStep; target network
    update steps: updateStep
    Output: Trained policy $\pi_{\zeta_a}$
1  Initialize replay buffer BF; $step \leftarrow 0$
2  for 1 to $M$ do
3      Input a sample into simulator to generate initial state $s_0$
4      for each time slot $T_j \in \mathcal{T}$ do
           // Introducing policy guidance stochastically.
5          randomValue $\leftarrow randomBetween(0, 1)$
6          if $randomValue \leq R^{PG}$ then
7              Get action $A(T_j)$ by rerunning the MILP solver:
                 $A(T_j) \leftarrow MILP(S(T_j), remainEpisode)$
8          else
9              Get masked action using current policy, actionNoise:
10             $A(T_j) \leftarrow Mask\left(S(T_j), \pi(S(T_j)|\zeta_a)) + actionNoise\right)$
11         State transition $S(T_{j+1}) \leftarrow Trans(S(T_j), A(T_j))$
12         Get the action reward $R(T_j) \leftarrow Reward(S(T_j), A(T_j))$
13         Store transition $(S(T_j), A(T_j), R(T_j), S(T_{j+1}))$ in BF
14         if step mod trainStep == 0 then
15             Sample batch $(S(T_i), A(T_i), R(T_i), S(T_{i+1})$ from BF
16             Get masked actions using target actor network:
17             $A(T_{i+1}) \leftarrow Mask(S(T_{i+1}), \pi'(S(T_{i+1})|\zeta'_a))$
18             Set target $y_i \leftarrow R(T_j) + \gamma Q'(S(T_{i+1}), A(T_{i+1})|\zeta'_c)$
19             Update critic network by minimizing the loss:
                 $L \leftarrow \frac{1}{N} \sum_i (y_i - Q(S(T_i), A(T_i)|\zeta_c))^2$
20             Get masked actions $A(T_i)$ at $S(T_i)$ using actor
               network: $A(T_i) \leftarrow Mask(S(T_i), \pi(S(T_i)|\zeta_a))$
21             Update the actor policy using policy gradient:
22             $\nabla_{\zeta_a} J \leftarrow$
                 $\frac{1}{N} \sum_i \nabla_a Q(S(T_i), A(T_i)|\zeta_c)|\nabla_{\zeta_a} \pi(s|\zeta_a)|_{S(T_i)}$
23         if step mod updateStep == 0 then
24             Update the target networks: $\zeta'_a \leftarrow \tau\zeta_a + (1 - \tau)\zeta'_a$;
                 $\zeta'_c \leftarrow \tau\zeta_c + (1 - \tau)\zeta'_c$
25         $step \leftarrow step + 1$
```

techniques. Action masking, denoted as $Mask(S(T_j), A(T_j))$, refines the raw actions generated by the actor network by enforcing action validity and utilizing domain-specific knowledge, thereby improving policy performance (lines 9, 14, 17). Meanwhile, policy guidance incorporates the MILP solver to provide optimal actions through $MILP(S(T_j), remainEpisode)$, based on current and future information (lines 5-9). These optimal actions are stochastically introduced during RL training into the replay buffer, mixing high-quality actions with the raw RL actions to enhance the training transition quality and guide the RL training in a beneficial direction.

A.4 Complementary Experimental Results

We evaluate our approach on an environment with the parameters shown in Table 8. We use the same parameters across hyperparameter tuning, training, evaluation, and comparison.

Additional Details on Hyper-Parameter Tuning: This methodology enables us to monitor and assess the RL model's performance throughout the training process. An early stopping procedure is implemented, terminating training if the total reward on the 20 evaluation samples does not improve after a specified number of iterations. Finally, we select the optimal combination of hyperparameters based on the results from the 3-fold cross-validation. The new RL model is then trained using the full set of training samples with the identified best parameter combination. Finally, we test the trained 9 RL models on 50 unseen monthly samples for each month from May 2023 to January 2024 to evaluate their generalization performance. Table 6 show selected values for each hyperparameter after tuning.

Table 6: Hyperparameters and selected values.

Parameter	Description	Range
Actor network	Number of units at each layer	[96, 96]
Critic network	Number of units at each layer	[96, 96]
Γ	Discount factor for future reward	1
Actor&Critic learning rate	Learning rate for updating actor and critic networks	10^{-5}, 10^{-3}
bufferSize	Batch size for fetching transitions from replay buffer	64
batchSize	Size of the replay buffer	10^6
actionNoise	Noise added during action exploration	normal(0,0.2)
R^{PG}	Probability to introduce policy guidance	0.5 or 0.7
$\lambda_S, \lambda_B, \lambda_D$	Penalty coefficients for SoC requirement, bill cost, and demand charge	1,1, 3
Random seed	Random seed for actor and critic network initialization	0-5
Adjustment of $\hat{p}^{max}$	Lower bound of the 99% confidence interval for the optimal monthly peak power based on training data	Increased by 0%, 5%, 10%
trainStep, *updateStep*	Training steps and steps per update of target networks	5, 5

```
Algorithm 3: Trickle Charging with Least Laxity First.
    Input: Set of EVs $V$, chargers $C$, time slots $t$
    Output: Charging schedule for each EV
for each time slot $T_j \in \mathcal{T}$ do
    for each EV $\phi(C_i, T_j)$ connected to chargers $C$ do
        Compute laxity: $L(\phi(C_i, T_j)) =$
          $(\mathcal{D}(\phi(C_i, T_j)) - t) - KWH^R(C_i, T_j)/C^i_{max}$.
    Compute power gap: $\hat{P} = \hat{P}^{max}(T_j) - \mathcal{B}(T_j)$;
    Sort EVs by laxity: $L(\phi(C_i, T_j))$ in ascending order;
    Initialize $[P(C_i, T_j)]_{C^i \in C}$ as 0;
    for each EV $\phi(C_i, T_j)$ sorted by laxity do
        if $\hat{P} > 0$ then
            Set trickle charging rate:
              $P(C_i, T_j) = \min(KWH^R(C_i, T_j)/\tau^R(C_i, T_j), \hat{P})$ ;
            $\hat{P} \leftarrow \hat{P} - P(C_i, T_j)$;
```

Heuristics: Algorithm 4 and Algorithm 3 show the exact algorithms used for Charge First LLF and Trickle LLF, respectively.

Ablation results: Finally, Table 3 shows a breakdown of the ablation results per month.

Table 7: Training and testing data information for each month.

Month	Training (60 Samples)					Testing (50 Samples)		
	Car Arrival Number (per day)	Monthly Peak Building Load (kW)	Daily Peak Building load (kW)	Estimated peak power (kW)	Number of Week-days	Car Arrival Number (per day)	Monthly Peak Building Load (kW)	Daily Peak Building Load (kW)
MAY	9.35 ± 2.19	125.89 ± 1.72	96.22 ± 13.76	119	22	9.31 ± 2.2	125.62 ± 1.65	96.18 ± 13.73
JUN	11.4 ± 2.53	141.04 ± 1.83	109.86 ± 14.0	125	21	11.49 ± 2.51	140.34 ± 2.21	109.75 ± 13.97
JUL	10.86 ± 2.45	148.08 ± 1.57	111.79 ± 35.39	145	20	10.97 ± 2.4	148.12 ± 1.72	111.94 ± 35.49
AUG	6.87 ± 1.97	221.02 ± 3.86	160.5 ± 27.92	202	23	6.92 ± 1.93	221.17 ± 3.77	160.69 ± 28.14
SEP	14.99 ± 2.9	145.91 ± 1.48	127.2 ± 11.86	143	20	14.94 ± 2.88	146.23 ± 1.08	127.2 ± 11.85
OCT	14.55 ± 2.91	186.54 ± 1.72	118.58 ± 30.12	174	21	14.54 ± 2.86	185.17 ± 2.77	118.5 ± 30.13
NOV	9.59 ± 2.31	144.9 ± 3.28	113.09 ± 19.18	130	19	9.57 ± 2.29	144.2 ± 2.92	113.09 ± 19.2
DEC	20.36 ± 3.4	116.49 ± 2.25	98.72 ± 8.0	104	16	20.32 ± 3.35	116.33 ± 2.01	98.69 ± 7.95
JAN	13.62 ± 2.73	137.77 ± 2.47	91.56 ± 19.46	127	21	13.53 ± 2.76	137.64 ± 2.5	91.72 ± 19.47

Table 8: Simulation Parameters.

Parameter	**Value**
C	15 chargers (5 bi-directional, 10 uni-directional)
C^i_{min}, C^i_{max}	[-20 kW, 20 kW] for bi-directional, [0, 20 kW] for uni-directional chargers
δ	Time interval: 0.25 hours
$\theta_D, \theta_E(T_j)$	9.62 $/kW (Demand), 0.11271 $/kWh (off-peak), 0.1466 $/kWh (peak: 6 a.m.-10 p.m.)
$CAP(V)$	EV battery capacity: 40 or 62 kWh
$SOC^{min}(V)$, $SOC^{max}(V)$	Minimum and maximum SoC: 0% and 90% of capacity

Table 9: Month-wise ablation results for total bill. Lower is better.

Policy	MAY	JUN	JULY	Total
RL	**6222.6**	**6857.0**	**7392.2**	**20471.0**
RL\500	6225.08	6875.12	7394.59	20,494.79
RL\E	6464.49	7033.71	7631.95	21,130.15
RL\A	6377.01	7174.84	7721.81	21,273.66
RL\C	6223.4	6895.8	7392.45	20,511.65
RL\F	6230.39	6968.64	7395.07	20,594.10
Random\A	6532.59	7250.97	7843.72	21,627.28

Algorithm 4: Charge First with Least Laxity First.

```
Input: Set of EVs $V$, chargers $C$, time slots $t$
Output: Charging schedule for each EV
for each time slot $T_j \in \mathcal{T}$ do
    for each EV $\phi(C_i, T_j)$ connected to $C$ do
        Compute laxity: $L(\phi(C_i, T_j)) = (\mathcal{D}(\phi(C_i, T_j)) - T_j) - KWH^R(C_i, T_j)/C^i_{max}$.
    Compute power gap: $\hat{P} = \hat{P}^{max}(T_j) - \mathcal{B}(T_j)$;
    Compute sum of trickle rates: $S = \sum_i KWH^R(C_i, T_j)/\tau^R(C_i, T_j)$;
    if $S < \hat{P}$ then // Overcharging EVs for future discharging
        for each EV $\phi(C_i, T_j)$ connected to $C$ do
            Set trickle rate: $P(C_i, T_j) \leftarrow KWH^R(C_i, T_j)/\tau^R(C_i, T_j)$;
        $\hat{P} \leftarrow \hat{P} - S$;
        for each EV $V = \phi(C_i, T_j)$ connected to bi-directional chargers in reverse laxity order do
            Charge to maximum SoC: $P(C_i, T_j) \leftarrow \min\left(C_i^{max}, \frac{(SOC^{max}(V) - SOC(V, T_j)) \times CAP(V)}{\Delta t}, \hat{P}\right)$;
            $\hat{P} \leftarrow \hat{P} - P(C_i, T_j)$;
    else // Discharging EVs to increase power gap
        while $\hat{P} \geq S$ and not all EVs connected to bi-directional chargers are considered do
            for each $V = \phi(C_i, T_j)$ connected to bi-directional chargers in reverse laxity order do
                if $SOC(V, T_j) > SOC^R(V)$ then
                    $P(C_i, T_j) \leftarrow \max\left(C_i^{min}, \frac{(SOC^R(V) - SOC(V, T_j)) \times CAP(V)}{\delta}\right)$;
                    $\hat{P} \leftarrow \hat{P} - P(C_i, T_j)$;
        for each $V = \phi(C_i, T_j)$ connected to chargers sorted by reverse laxity do
            Set trickle rate: $P(C_i, T_j) \leftarrow \min\left(KWH^R(C_i, T_j)/\tau^R(C_i, T_j), \hat{P}\right)$;
            $\hat{P} \leftarrow \hat{P} - P(C_i, T_j)$;
```