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

Fangqi Liu
Rishav Sen
Jose Paolo Talusan
fangqi.liu@vanderbilt.edu
rishav.sen@vanderbilt.edu
jose.paolo.talusan@vanderbilt.edu
Vanderbilt University
Nashville, TN, USA

Ava Pettet
Aaron Kandel
Yoshinori Suzue
ava.pettet@nissan-usa.com
aaron.kandel@nissan-usa.com
yoshinori.suzue@nissan-usa.com
Nissan Advanced Technology Center -
Silicon Valley
Santa Clara, CA, USA

Ayan Mukhopadhyay
Abhishek Dubey
ayan.mukhopadhyay@vanderbilt.edu
abhishek.dubey@vanderbilt.edu
Vanderbilt University
Nashville, TN, USA

ABSTRACT

Strategic aggregation of electric vehicle batteries as energy reservoirs can optimize power grid demand, benefiting smart and connected communities, especially large office buildings that offer workplace charging. This involves optimizing charging and discharging to reduce peak energy costs and net peak demand, monitored over extended periods (e.g., a month), which involves making sequential decisions under uncertainty and delayed and sparse rewards, a continuous action space, and the complexity of ensuring generalization across diverse conditions. Existing 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 [25]. 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 [26, 31] 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 [27, 29]. 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 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

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

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

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 major EV manufacturer with access to a smart building that has 15 heterogeneous chargers (Figure 1 shows some of them). We use real-world charging and energy data to validate our approach, showing its effectiveness in reducing energy costs over nine 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 novel RL 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 conduct a detailed ablation study to assess the impact of each technique and demonstrate the model's effectiveness.

2 PROBLEM FORMULATION

Charger and Time Intervals: Consider the building has N heterogeneous chargers $\mathcal{C} = \{C_1, C_2, \ldots, C_N\}$. 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 $\mathcal{T} = \{T_1, T_2, \ldots T_{end}\}$, where $T_j - T_{j-1} = \delta$ (we use δ = 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 $\mathcal{P} : \mathcal{C} \times \mathcal{T} \rightarrow \mathfrak{R}$ 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 $\mathcal{B} : \mathcal{T} \rightarrow \mathfrak{R}^+$ 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 $\theta_E : \mathcal{T} \rightarrow \mathfrak{R}^+$ (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 $\mathcal{P} = \{P(C_i, T_j) | C_i \in \mathcal{C}, T_j \in \mathcal{T}\}$.

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 denote 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 $SOC^I : \mathcal{V} \rightarrow \mathfrak{R}^+$ and the required final state of charge (measured as a percentage of the battery capacity) $SOC^R : \mathcal{V} \rightarrow \mathfrak{R}^+$ upon arrival. $SOC^{min} : \mathcal{V} \rightarrow \mathfrak{R}^+$ is the minimum allowed SoC for the car i.e., the car cannot be discharged below this value, and $SOC^{max} : \mathcal{V} \rightarrow \mathfrak{R}^+$ 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: Our approach employs a two-layer decision-making process for EV charging optimization. First, a heuristic assigns EVs to chargers upon arrival. Second, an RL-based policy optimizes charging rates atfi xed intervals. 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 $\phi(C_i, T_j) = \emptyset$. This underscores the dynamic nature of charger assignments, which ensures that no two electric vehicles share a charger simultaneously. Our FIFO policy prioritizes bidirectional chargers as the optimal strategy (see Table 4 in the appendix[1]), enhancing charging efficiency. We also maintain the connection between the assigned charger and the EV until departure. For EV charging, we approximate a linear charging profile, following prior work [23]. The SoC is updated at each 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 charges, 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 3 of the appendix.*

Uncertainty of vehicles and SoC requirements. Meta-heuristics and Model Predictive Control (MPC) have 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 [27]. 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. V2B optimization is difficult due to long billing periods. While prior work (barring some exceptions [8]) optimizes and plans 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, 30], 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 approaches using simulations with limited interface with the real world (barring some exceptions [8]), thereby making simplistic assumptions that limit deployment.

4 OUR APPROACH

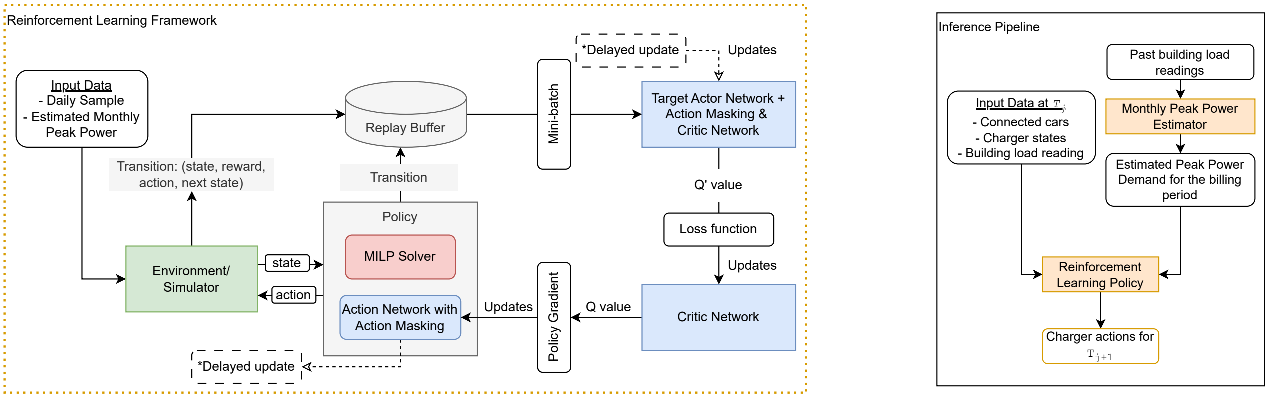
In this section, we discuss the different components in our framework, shown in Figure 2a.

4.1 Markov Decision Process Model

We model the V2B problem as the following MDP.

State. The complete state space for the problem can be described using features that capture 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, 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.

[1] The full paper, including the appendix, is available on arXiv.


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

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 $[KWH^R(C_i, T_j)]_{C_i \in C}$, which is initialized to 0. This quantity represents the energy gap between required SoC (SOC^R) and current SoC (SOC) of the EV $V = \phi(C_i, T_j)$, defined as $KWH^R(C_i, T_j) = (SOC^R(V) - SOC(V, T_j)) \times$ 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 $T_j \in \mathcal{T}$. 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 $\eta(V)$ byfirst 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 $Reward : \mathcal{S} \times \mathcal{A} \longrightarrow \Re$, 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.

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.

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 $toImprove \leftarrow \min(ReLU(powerGap - \sum A'), \sum canIncrease)$
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 $A' \leftarrow A' + \frac{toImprove \times negAction}{\sum(negAction) + \epsilon}$
```

4.2.1 *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 afi xed sample, i.e., afi xed 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. Thefi xed 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] as a policy guidance subroutine [10] in the RL training process. The solver, given the current state and future events, provides optimal charging actions. Each training dataset contains complete episode data, enabling the MILP solver to account for future dynamics. During RL training, it generates optimal actions based on the current state and full future information of the episode (i.e., a full-month billing period). The solver is stochastically triggered, and its outputs are added to the replay buffer with a predefined coefficient, R^{PG} (see Algorithm 2 in the appendix). The next optimal action is computed as $MILP(S(T_j), remainEpisode)$, considering factors such as EV arrivals, SoC requirements, and building power. By blending MILP-generated actions with those from the RL actor network, the agent explores a more effective action space, improving its ability to handle large continuous action spaces and long-term rewards.

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) 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 Transmission System Operators 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 collected 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 powerfluctuations, 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 7 (appendix). We sampled 1000 billing episodes for each month.

Downsampling. We found that increasing training samples beyond a certain limit raised computational demands and worsened performance (see ablation study in Section 5.2). To address this, we applied k-means clustering [5] with $k = 5$, using optimal demand charges from the MILP solution to select 60 training samples and 50 testing samples per cluster, ensuring exclusivity. As shown in Table 6 (appendix), the training and testing datasets span nine months, capturing variations in daily EV arrivals, peak building loads. Daily arrivals range from 6.87 (August) to 20.36 (December), reflecting seasonal demand shifts, while monthly peak building loads vary from 116.49 kW (December) to 221.02 kW (August), demonstrating diverse energy consumption patterns affecting charging strategies.

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 5 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. We transform training data into input samples for our digital twin/simulator, Optimus [24], which simulates the EV charging scenario. To evaluate our RL approach, we compare it with an optimal oracle, a real-world charging baseline, and several heuristics. Brief baseline descriptions are provided here, with details 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 [28]. 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. Wefi rst 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) **Total Bill:** The sum of electricity cost and demand charge over the billing period, computed by Eq. (6) and 2) **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. Additionally, missing SoC—the energy shortfall between required and actual SoC at departure—is critical in the V2B problem. Our RL model, with action masking, ensures all EVs reach their required SoC before departure by applying force charging and discharging in **Mask 2** and **Mask 3**. For fairness, these force procedures are applied across all proposed heuristics, effectively minimizing missing SoC. Therefore, we do not report this metric separately.

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. 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 in balancing the charging and discharging process, offering better overall performance across all heuristics. Table 9 in Appendix A.4 illustrates the peak shaving performance across all approaches, showing that our RL approach achieved peak shaving in six months (indicated by positive values), demonstrating its effectiveness in reducing demand charges through EV charging.

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

Table 1: Total Bill on Test Set (Lower is Better). Best Values in Bold. MILP Provides the Optimal Solution with Oracle Input. (Peak Shaving Results is shown in Table 9 in the Appendix.)

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 (Ours)	**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: Ablation Results for the Total Bill Over Three Months (Lower is Better).

RL (Ours)	**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

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 2 and Appendix A.3.

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.

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. 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.

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.

Training without the 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.

To assess the impact of the actor network, we replaced it with a random policy in the **Random\A** approach, where random charging actions are generated before applying action masking. Its poor performance highlights that action masking alone is insufficient, emphasizing the actor network's critical role in achieving optimal outcomes. While all proposed heuristics (except FC and Trickle) adhere to action masking constraints, including forced charging and power allocation based on estimated peak power, the RL approach consistently outperforms them, reinforcing 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 heterogeneous (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 heuristic algorithms in simulated V2B scenarios with real-world data from an EV manufacturer. Results show that our trained models effectively manage online EV charging, reducing 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] Jose Paolo Talusan, Rishav Sen, Ava Pettet, Aaron Kandel, Yoshinori Suzue, Liam Pedersen, Ayan Mukhopadhyay, and Abhishek Dubey. 2024. OPTIMUS: Discrete Event Simulator for Vehicle-to-Building Charging Optimization . In *2024 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE Computer Society, 223–230. https://doi.org/10.1109/SMARTCOMP61445.2024.00050

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

[26] 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

[27] 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

[28] 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.

[29] 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.

[30] 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

[31] 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