AI Engineering
https://www.sciltp.com/journals/aieng

Article

Check for updates

Adaptive Multi-Agent Reinforcement Learning for Electric Vehicle On-Road Charging Optimization Under Dynamic Traffic Conditions

Shaghayegh Rabbanian [1], Hao Wang [2,*] and Lei Wu [2]

[1] Division of Computer Science and Engineering, Louisiana State University, Baton Rouge, LA 70803, USA
[2] Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA
* Correspondence: hwang9@stevens.edu

How To Cite: Rabbanian, S.; Wang, H.; Wu, L. Adaptive Multi-Agent Reinforcement Learning for Electric Vehicle On-Road Charging Optimization Under Dynamic Traffic Conditions. *AI Engineering* **2026**, *2*(1), 2. https://doi.org/10.53941/aieng.2026.100002

Received: 23 September 2025
Revised: 12 April 2026
Accepted: 14 April 2026
Published: 9 May 2026

Abstract: Given the rapid growth of Electric vehicles (EVs) and their finite battery life, improving on-road charging strategies presents significant challenges, including customer dissatisfaction due to long waiting times and the dynamic nature of the traffic data can lead to choosing suboptimal charging station. Traditional approaches primarily depend on fixed road data; our model utilizes real-time traffic data from Google Maps to capture live congestion patterns and dynamically optimize charging assignments for the vehicles to minimize costs and improve overall driver satisfaction. This study focuses on three fundamental questions for the optimal on-road charging of a fleet of EVs: (i) when is the best time to charge the vehicle; (ii) where is the optimal charging location for each EV; and (iii) how should charging be planned considering the condition of the road and battery level. Our objective is to determine the optimal time and location for electric vehicle charging that minimizes the weighted sum of travel and charging time, charging cost, and associated penalties for late charging and station overutilization, while taking into account key factors such as real-time traffic conditions, the spatial distribution of charging stations, and EV-specific attributes such as state of charge (SOC), driving range, and travel efficiency. To develop a robust and adaptive EV charging recommendation system, we employ Multi-Agent Reinforcement Learning (MARL) to derive an intelligent, self-improving charging strategy that dynamically adapts to changing situations. Numerical simulations demonstrate that our model provides an applicable and scalable solution for EV users and urban planners, contributing to more efficient and intelligent EV charging infrastructure. We also compared the results of our MARL model with an exact approach in small-scale using Gurobi package in Python.

Keywords: electric vehicle; charging recommendation; multi-agent reinforcement learning

1. Introduction

Electric vehicles (EVs) have become increasingly popular in modern transportation due to their minimal greenhouse gas emissions and high energy efficiency, addressing heightened environmental and energy concerns. Many countries are promoting EVs as an eco-friendly alternative to fossil-fuel vehicles. Their clean-energy profile advocates environmental sustainability, while their lower operating costs and convenient driving experience present them as a favorable choice for buyers. According to the International Energy Agency (IEA), the global EV market has experienced rapid growth, with over 7 million EVs on the road as of 2020 and projections exceeding 250 million by 2030 [1]. This rapid growth—fueled by government incentives, advances in battery technology, and increasing environmental awareness—has led to a rising demand for effective and reliable charging infrastructure.

Despite efforts to expand charging networks, EV drivers still encounter long queues and a limited number of charging stations. These issues increase EV charging costs, reduce efficiency, and intensify range anxiety, eventually

diminishing user satisfaction and impeding broader adoption. To mitigate these problems, intelligent charging recommendation systems [2,3] have been proposed. These systems guide drivers to optimal charging stations based on factors such as station availability, proximity, and travel destination, thereby improving charging convenience and overall system efficiency.

Two primary factors influence EV drivers' charging decisions: the availability of the charging stations and traffic conditions. Integrating real-time multidimensional data from EVs, traffic systems, and charging infrastructure is essential for developing an efficient fast-charging guidance strategy. This approach is key to alleviating EV drivers' range anxiety and minimizing the inconvenience associated with charging, two major challenges that currently hinder the broader adoption of EVs [4].

Research problem. A key challenge for on-road charging guidance is that the system is inherently dynamic: travel times vary due to congestion and incidents, station usage changes over time, and EVs differ in SOC and energy efficiency. As a result, a charging plan that is optimal at departure may become suboptimal after unexpected disruptions, potentially causing additional detours, longer queues, and even infeasible trips. Therefore, effective charging coordination requires sequential and adaptive decision-making rather than one-shot planning.

Our study aims to improve user satisfaction in EV charging systems by integrating dynamic reassignment mechanisms that adapt to real-time disruptions (e.g., traffic accidents) which can render initial charging plans suboptimal. The system can transfer EVs to different charging stations, thereby reducing the total time spent traveling and charging by constantly checking the availability of charging stations and travel conditions. This dynamic reassignment not only reduces the time users have to wait and the energy used for traveling, but also decreases charging costs by sending EVs to stations that are less occupied or more cost-effective. Adding this kind of real-time response to the recommendation system makes the charging experience more stable and user-friendly, especially when traffic conditions vary significantly.

To effectively manage the complex dynamic nature of EV charging coordination under real-world conditions, we employ a Multi-Agent Reinforcement Learning (MARL) framework. Unlike traditional static optimization approaches, MARL enables individual EVs to act as autonomous agents that dynamically learn optimal charging decisions through interaction with the environment. In MARL, the decentralized approach enables the system to adapt to real-time changes such as traffic accidents, fluctuating station loads and energy prices, ultimately improving efficiency and user satisfaction. Additionally, MARL helps with continuous learning, allowing the system to generalize across different scenarios and respond to unexpected disruptions. By simulating diverse environments during training, the agents can develop robust policies that perform well even when the initial plan becomes suboptimal.

Objectives. This paper aims to develop an adaptive on-road charging coordination framework for an EV fleet under dynamic traffic conditions. Specifically, we target three fundamental questions: (i) when to charge; (ii) where to charge; and (iii) how to adjust charging decisions when traffic conditions change. Our specific objectives are to: (1) formulate multi-EV on-road charging coordination with time-varying traffic and station congestion as a Markov game suitable for multi-agent learning; (2) design a MARL environment that enforces physical feasibility (e.g., SOC reachability and station capacity) while enabling real-time reassignment under disruptions; and (3) learn decentralized charging-station selection policies using a Centralized Training Decentralized Execution (CTDE)-based MARL algorithm, namely Multi-Agent Proximal Policy Optimization (MAPPO), and evaluate performance under both static and dynamic conditions.

Contributions. Building upon prior work on MARL-based EV charging recommendation and coordination [2–4], the proposed solution introduces the following contributions:

- **Dynamic, disruption-aware coordination**: Learning decentralized EV charging and reassignment policies that respond to time-varying traffic conditions and incident-driven disruptions while considering station congestion and EV feasibility constraints.
- **Feasibility-aware MARL environment design**: Integrating reachability and operational constraints (e.g., SOC-based station reachability, charging eligibility, and station capacity/overload handling) via state features and action masking to ensure realistic learning and execution.
- **Benchmarking against an exact optimization baseline**: Comparing the learned MARL policy with a Gurobi-based exact optimization model in small-scale settings to validate solution quality and to illustrate the value of adaptivity under dynamic scenarios.

2. Literature Review

2.1. MARL Foundations

Multi-agent Reinforcement Learning (MARL) extends Single-Agent Reinforcement Learning (SARL) to problems involving multiple autonomous agents acting simultaneously, typically modeled as stochastic games [5].

As Figure 1 shows, agents interact with both the environment and each other, where each agent has its own action space, observation, and reward, while the joint policy determines the global outcome. MARL introduces challenges such as non-stationarity—the "moving target" problem where changing policies of other agents affect each agent's perceived environment [6]—making convergence difficult in highly interactive systems.

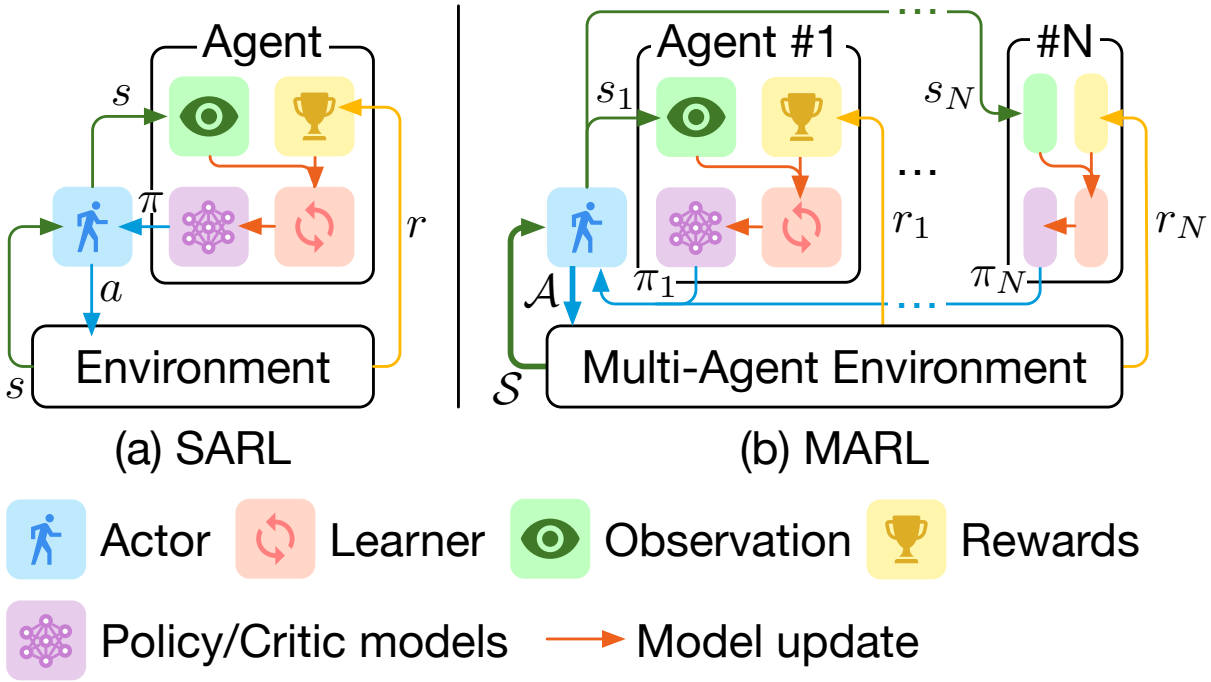

Figure 1. SARL v.s. MARL. (**a**) SARL only employs one agent to interact with the environment by learning a policy $\pi(a \mid s)$, which indicates taking action a given the state s; (**b**) MARL involves multiple agents interacting with the environment and each other, fostering a stochastic game characterized by joint state and action spaces, $\mathcal{S}$ and $\mathcal{A} = \prod_{i=1}^{N} \mathcal{A}_i$, as well as shared or individual rewards r_i^t. Agents learn their own policies $\pi_i(a_i|s_i)$ for $\forall i \in \{1, \ldots, N\}$, with individual partial observation s_i [7].

Centralized Training with Decentralized Execution (CTDE) addresses coordination and partial observability challenges by training agents with full state information while executing with only local observations [8]. In actor-critic architectures, a centralized critic uses global information during training, while decentralized actors make decisions using local inputs during execution [9,10]. This enables sample-efficient learning and coordination while maintaining scalability. Among CTDE algorithms, Multi-Agent Proximal Policy Optimization (MAPPO) [11] employs clipped surrogate objectives for stable policy updates, making it well-suited for cooperative multi-agent problems like EV charging coordination.

In this work, we leverage MAPPO's stability and scalability to learn disruption-aware coordination policies that dynamically respond to traffic incidents and station congestion. Unlike many CTDE-based applications that assume static or slowly varying environments, our approach explicitly embeds feasibility constraints into the MARL environment via action masking and feasibility-aware state features, ensuring that the learned policies remain executable under operational limits such as SOC-based reachability and station capacity. Moreover, we model charging decisions jointly with en-route travel conditions (i.e., routing-dependent travel time to candidate stations), rather than treating routing and charging as separate stages or assuming a predetermined route.

2.2. Applications to EV Routing and Charging

Recent MARL applications to EV charging address fluctuating electricity prices, queuing dynamics, and real-time traffic disruptions. RL-CHARGE [12] employs CTDE with spatiotemporal heterogeneous graph convolution networks to model station interactions and reduce waiting time, cost, and failure rates in urban environments. Li et al. [13] proposed a dual-timescale framework combining slow-timescale electricity cost optimization with fast-timescale charging recommendations using graph attention networks to capture station competition and routing dynamics. Zhang et al. [14] applied CommNet to residential EV charging, balancing user needs with grid stability, though scalability challenges remain. Suanpang et al. [15] demonstrated 20% efficiency improvement and 15% cost reduction in Thailand's dynamic charging networks by adapting to real-time demand and price fluctuations.

While MARL offers robustness, parallelism, and scalability [5], challenges include the curse of dimensionality

and the need for careful reward shaping. Nonetheless, MARL's ability to handle cooperative behavior and dynamic conditions—including real-time incidents—makes it well-suited for modern EV charging coordination under practical constraints.

This work distinguishes itself from prior studies by explicitly incorporating real-time traffic disruptions (e.g., accidents that alter travel times) into the MARL framework, enabling dynamic reassignment decisions that adapt to evolving conditions during execution. Furthermore, by benchmarking the learned MAPPO policy against a Gurobi-based exact optimization baseline under deterministic scenarios, we validate solution quality while demonstrating the value of learning-based adaptivity when disruptions occur.

3. Mathematical Model

3.1. Optimization Model Formulation

Using the indices, parameters, and decision variables in Table 1, we present a mathematical model. For the proposed EV on-road charging optimization framework, the objective function is shown in Equation (1), which minimizes the total cost associated with EV charging decisions by considering time, monetary, and operational efficiency factors. This objective setup ensures that the charging schedule is both cost-efficient and operationally realistic by balancing travel and wait times, energy costs, and charger availability.

Table 1. Indices, parameters, and variables of the model.

Indices	
$d = 1, \ldots, D$	Index of destinations for EVs
$i = 1, \ldots, I$	Index of EVs
$j = 1, \ldots, J$	Index of charging stations
Parameters	
α_1	Weight applied to time-based components (travel, wait, charging)
α_2	Weight applied to monetary cost (charging price, energy, time value)
CU_j	Number of charging units at station j
D_i	Destination location of EV i
d_{id}	Direct distance from V_i to D_i when no charging is involved (miles)
d_{ij}	Distance from O_i to S_j based on Google Maps data (miles)
d_{jd}	Distance from S_j to D_i based on Google Maps data (miles)
E_i	Energy consumption rate of EV i (MWh/mile)
E_i^{cap}	Total battery capacity of EV i (kWh)
E_i^{min}	Minimum SOC required at the destination for EV i
ϵ	Small delay penalty to favor earlier time steps
$\mathbb{1}_{\mathrm{overload}}$	Indicator function set to 1 if the chosen station is overloaded
O_i	Origin location of EV i
P_j	Charging power at station j (kW)
β	Queue penalty coefficient
λ	Small constant encouraging better load balancing
R_j	Charging cost per kWh at station j ($/kWh)
S_j	Location of charging station j
SOC_i	Initial SOC of EV i
T_{id}	Direct travel time from O_i to D_i when no charging is involved (hours)
T_{jd}	Travel time from S_j to D_i based on Google Maps data (hours)
T_{ij}	Travel time from O_i to S_j based on Google Maps data (hours)
t_i^{depart}	Departure time of EV i from origin
TEC_j	Time equivalent cost at station j (hour/$)
Δt	Time duration per step (hours)
Variables	
δ_i	Equal to 1 if EV i is charged during the trip, 0 otherwise
$E_{ijt}^{\mathrm{charge}}$	Energy charged by EV i at station j at time t (kWh)
O_{jt}	Overload amount at station j at time t
$T_{ijt}^{\mathrm{charge}}$	Charging time for EV i at station j at time t
x_{ijt}	Equal to 1 if EV i is assigned to station j at time t, 0 otherwise

$$\begin{aligned}\min \quad & \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\Big\{x_{ijt}\cdot\Big[\alpha_1\cdot(T_{ij}+T_{ijt}^{\text{charge}}+T_{ji}) \\ & +\alpha_2\cdot(R_j\cdot E_{ijt}^{\text{charge}}\cdot \text{TEC}_j)+\epsilon\cdot t\Big]\Big\} \\ & +\lambda\cdot\sum_{j=1}^{J}\sum_{t=1}^{T}O_{jt},\end{aligned} \tag{1}$$

The objective function (1) includes four terms: (i) The first term incorporates three time-related components, including the travel time T_{ij} from the origin to the selected charging station, the time T_{ijt}^{charge} spent on charging, and the travel time T_{jd} from the station to the final destination. These three components are weighted by a time preference coefficient α_1, ensuring that routes and charging schedules with shorter durations are favored; (ii) The second term represents the charging cost, which depends on the energy charged at each station E_{ijt}^{charge}, the per-unit energy cost R_j, and a time-equivalent multiplier TEC_j that converts monetary cost into time-equivalent units. This term is weighted by a cost preference coefficient α_2, allowing for trade-offs between time efficiency and monetary cost; (iii) To promote earlier charging when feasible, the third term representing a small penalty $\epsilon \cdot t$ is added, where t denotes the time index. This encourages the model to prioritize assignments at earlier time intervals when it is optimal to do so; (iv) The fourth term includes penalization on the station overutilization through auxiliary variables O_{jt}, which quantify excess demand beyond the number of available chargers CU_j at station j of time t. A small weight is assigned to this term to discourage assignments that would exceed station capacity without making the model infeasible. The first three terms are multiplied by variable x_{ijt}, ensuring that they are only counted if EV i is assigned to station j at time t.

The prevailing constraints are described as follows:

Equation (2) calculates the amount of energy EV i charges at station j at time t, which should be no larger than the energy capacity of EV i:

$$E_{ijt}^{\text{charge}} = P_j \cdot T_{ijt}^{\text{charge}}; 0 \le E_{ijt}^{\text{charge}} \le E_i^{cap} \cdot x_{ijt}, \ \ \forall i,j,t \tag{2}$$

Equation (3) determines if charging is required for each EV, while assuming that each EV will be charged at most once during the full trip. Specifically, an EV does not need to charge (i.e., $\delta_i = 0$) if it has enough SOC to reach the destination and maintain at least $E_i^{\min}$; an EV must charge (i.e., $\delta_i = 1$) if it cannot reach the destination with at least $E_i^{\min}$:

$$\sum_{j=1}^{J}\sum_{t=1}^{T} x_{ijt} = \delta_i, \quad \forall i \tag{3a}$$

$$\begin{aligned}-B\cdot\delta_i \le \text{SOC}_i\cdot E_i^{\text{cap}} - E_i\cdot d_{id} - E_i^{\min} \\ \le B\cdot(1-\delta_i), \quad \forall i\end{aligned} \tag{3b}$$

Equation (4) describes that, if an EV needs to be charged, it must be able to reach the assigned charging station before running out of energy.

$$\text{SOC}_i\cdot E_i^{\text{cap}} - E_i\cdot d_{ij}\cdot x_{ijt} \ge 0, \quad \forall i,j,t \tag{4}$$

Equation (5) expresses that if an EV is assigned to charge at a station, it will be adequately charged so that its SOC is at least $E_i^{\min}$ when reaching the destination. This prevents the situation where the EV charges too early in a station but does not have enough remaining energy to meet the minimum energy requirement at the destination.

$$\begin{aligned}\text{SOC}_i\cdot E_i^{\text{cap}} + x_{ijt}\cdot(-E_i\cdot d_{ij} + E_{ijt}^{\text{charge}} - E_i\cdot d_{jd}) \\ \ge E_i^{\min}, \quad \forall i,j,t\end{aligned} \tag{5}$$

Equation (6) presents the capacity limit of each EV.

$$\text{SOC}_i\cdot E_i^{\text{cap}} + x_{ijt}\cdot(-E_i\cdot d_{ij} + E_{ijt}^{\text{charge}}) \le E_i^{\text{cap}}, \quad \forall i,j,t \tag{6}$$

Constraint (7) ensures that EV i can only begin charging at station j in time t if it has already arrived there, i.e., t must be no earlier than the discretized arrival time at station j. This avoids assigning a charge before the EV physically arrives at the station.

$$x_{ijt} = 0, \quad \forall i, j, t < \left\lfloor \frac{T_{ij}}{\Delta t} \right\rfloor \tag{7}$$

Constraint (8) ensures that the overload variable O_{jt} captures the number of EVs assigned to station j at time t beyond its capacity CU_j. If the number of assigned EVs exceeds the available capacity, the excess is captured through O_{jt}; otherwise, O_{jt} remains zero as a result of the minimization setup.

$$O_{jt} \geq \sum_{i=1}^{I} x_{ijt} - \text{CU}_j, \quad \forall j, t \tag{8}$$

Constraint (9) ensures that the number of EVs being charged at station j at time t does not exceed the station's available number of charging units CU_j. It enforces a hard limit on concurrent charging.

$$\sum_{i=1}^{I} x_{ijt} \leq \text{CU}_j, \quad \forall j, t \tag{9}$$

3.2. Euclidean Distance Calculation

To compute the distance between two points on the Earth's surface using their latitude and longitude, we employ the Euclidean distance formula (10), where D represents the Euclidean distance in degrees, ϕ_i is the latitude of point i, and λ_i is the longitude of point i.

$$D := \sqrt{(\phi_1 - \phi_2)^2 + (\lambda_1 - \lambda_2)^2} \tag{10}$$

To convert this distance from degrees to kilometers, the result is multiplied by 111,045 m, as one degree on Earth is approximately equal to 69 miles or 111,045 meters. This method serves as a heuristic for estimating straight-line distances unaffected by obstacles (representing a straight-line distance) [16]. To estimate travel time, we assume that all EVs move at a constant speed. In scenarios without accidents, the travel time is calculated by dividing the distance by the assumed speed. In contrast, when an accident occurs, the travel time for the affected route is adjusted by multiplying it by a severity-dependent factor to reflect the resulting delay.

4. Markov Decision Process (MDP) Formulation and Environment for MARL

4.1. MDP Formulation

The mathematical problem presented in Section 3 is formulated as an MDP, which can be represented as a five-tuple (S, A, P, R, γ). S is the state space describing the system at a given time step; A is the set of feasible actions available to each EV; P is the state transition probability function; R is the reward function evaluating the quality of an action; γ is the discount factor balancing immediate and future rewards. Figure 2 illustrates the neural network architecture used by each MARL agent. The network takes agent-level state features (e.g., state of charge, reachability to charging stations, travel times, and station usage) as input, processes them through a single fully connected hidden layer with 64 neurons and rectified linear unit (ReLU) activation, and outputs Q-values corresponding to candidate charging station actions. This diagram provides a clear visualization of how agent observations are mapped to decision values during training.

State Space: At time t, the state of each EV agent i is represented as:

$$S_t^i = \left\{ \begin{matrix} \text{reachable}_{ij},\ \text{arrival}_{ij},\ \text{usage}_j,\ R_j, \\ P_j,\ \text{TEC}_j,\ \text{SOC}_i,\ \delta_i,\ \textit{accident} \end{matrix} \right\},$$

where reachable_{ij} is a binary vector indicating whether station j is reachable based on the current SOC and location of EV i; arrival_{ij} is arrival time of EV i at station j; usage_j is the current usage ratio of station j at time t; R_j is charging cost per kWh at station j; P_j is charging power of station j (kWh/hour); TEC_j is time-equivalent cost at station j (hours/$); SOC_i is the current SOC of EV i; δ_i is binary variable indicating whether EV i has already charged; *accident* is a binary variable indicating the presence of an accident along the route from the origin to a charging station for EVs that still require charging.

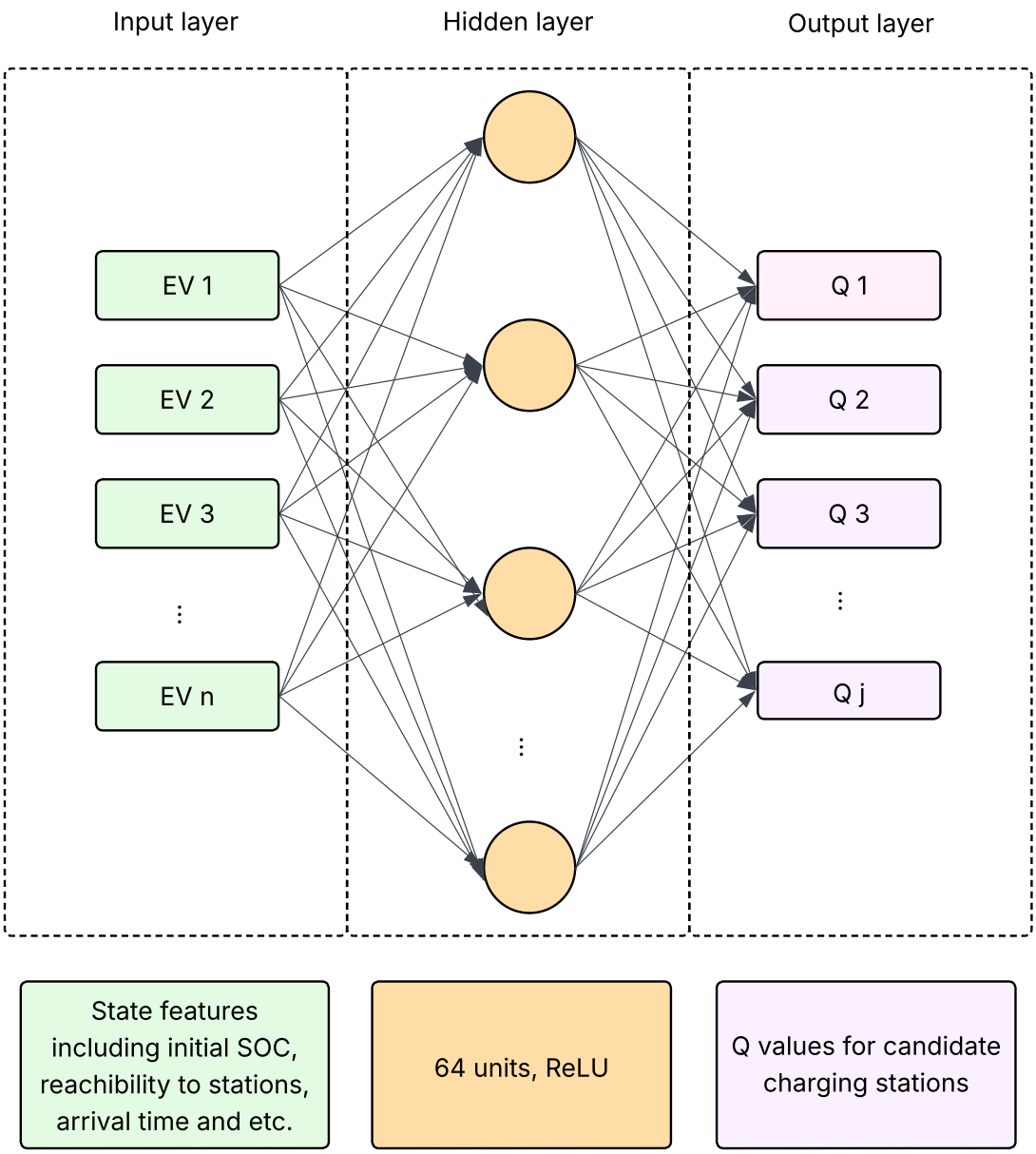

Figure 2. Neural network architecture of each MARL agent. Agent-level state features are passed through a fully connected hidden layer with 64 neurons and ReLU activation to produce *Q*-values for each candidate charging station action.

Action Space: At each time t, each EV agent i selects an action from a discrete set containing all available charging stations and a no-op (no action) option. The action space is defined as: $A_t^i \in \{0, 1, 2, \ldots, J\}$, where J is the total number of charging stations. Each action can take the following values:

- $A_t^i = 0$: EV i chooses to take no action at time t, i.e., it skips charging. This option is only allowed when the EV does not need to charge.
- $A_t^i = j$ with $j \in \{1, 2, \ldots, J\}$: EV i selects charging station j at time t and initiates charging if the station is reachable and available. The EV's energy level, travel distance, and station availability are used to determine whether this action is valid.

The environment automatically filters out infeasible actions at each time step using a binary action mask. While reachable_{ij} is a feature exposed to the policy network, the action mask is a hard constraint used by the environment, ensuring that EVs can only select charging stations that are reachable given their current SOC levels and locations, and prevents the no-op action when an EV is required to charge (i.e., when it does not have enough energy to reach its destination without charging). By combining reachable stations and conditional no-op allowance, the action space design supports both flexibility and policy enforcement to ensure that agents learn realistic and efficient behaviors. If an unreachable charging station was selected (for example, during early exploration when the mask might not have been strictly enforced), the environment will prevent the vehicle from executing that move.

The action is nullified and the agent receives a penalty (negative reward) for that time step. This penalty feedback signals the agent that the choice was invalid. During training, the reinforcement learning algorithm uses this negative reward to update the policy network, thereby learning to avoid such infeasible actions in the future.

To reduce the action searchspace, the action mask enforced by the environment will reduce the effective action space at any given decision point. Applying hierarchical decision-making [17,18] and function approximation for actions [19,20] can further mitigate the curse of dimensionality.

Transition Probability: The transition probability $P(S_{t+1}^i \mid S_t^i, A_t^i)$ defines how the environment evolves from one state to the next based on the agent's action. In this model, state transitions are influenced by the charging related decisions, the EV's current SOC, and the dynamic travel environment.

- *Charging at Station j*: If EV i selects action $A_t^i = j$ for $j \in \{1, \ldots, J\}$, the EV is assigned to charging station j at time t. The SOC is updated as: $\text{SOC}_i^{t+1} = \text{SOC}_i^t - E_i \cdot d_{ij}/E_i^{cap} + \min\left(P_j \cdot \Delta t, E_i^{\text{cap}} - \text{SOC}_i^t \cdot /E_i^{cap} + E_i \cdot d_{ij}\right)/E_i^{cap}$. The formula accounts for energy consumed to reach the station and the energy gained during charging, constrained by the battery's remaining capacity.
- *Skipping Charging (no-op)*: If EV i selects the no-op action $A_t^i = 0$, it does not initiate charging and continues to travel. Its SOC is updated as: $\text{SOC}_i^{t+1} = \text{SOC}_i^t - E_i \cdot d_{id}/E_i^{cap}$, capturing the energy depletion resulting from continued movement without recharging.
- *Temporal Dynamics*: The state also incorporates temporal changes. Travel time values T_{ij} and T_{jd}, along with station congestion (queue length), dynamically evolve based on spatial distances and past actions. These values are derived from the Euclidean-based distance and average travel speed and are used to simulate real-world uncertainty (e.g., traffic delays or queue buildup).
- *Station Usage Effects*: Station-level attributes, such as charging unit availability and time-specific station load, influence the system's evolution. If a station is overloaded, queuing penalties are applied to indicate that the agents may experience longer effective delays or reduced access to charging when selecting this station.

Overall, the transition dynamics are non-deterministic due to the interaction of agent decisions, resource constraints, and spatial-temporal variables, making this problem well-suited for learning-based approaches such as MARL.

Reward Function: The reward function is designed to guide individual EV agents toward minimizing the overall costs of charging and travel, balancing time efficiency, monetary cost, and system congestion. At each time t, the reward assigned to EV i is:

$$R_i^t = \underbrace{\alpha_1 \cdot \left(T_{ij} + T_{ijt}^{\text{charge}} + T_{jd}\right)}_{\text{travel time cost}} + \underbrace{\alpha_2 \cdot \left(R_j \cdot E_{ijt}^{\text{charge}} \cdot \text{TEC}_j\right)}_{\text{charging cost}} + \underbrace{\epsilon \cdot \Delta t + \lambda \cdot \mathbb{1}_{\text{overload}}}_{\text{delay penalty}}, \quad \forall j \tag{11}$$

The total cost includes three major components: (i) Travel and charging Time weighted by α_1; (ii) Charging cost scaled by both energy charged and the station's time-equivalent value; and (iii) Delay penalty applied proportionally to the elapsed time. The queue penalty is intended to capture the disutility or cost associated with an EV arriving at a charging station that already has a queue (i.e., other EVs waiting for service). We quantify this penalty in terms of estimated waiting time or queue length at the station. Specifically, for each station, we define an "overload" indicator based on the station's capacity and current number of vehicles waiting. If an EV selects an overloaded station (i.e., the number of EVs charging exceeds its available units), an additional penalty is added to discourage such actions. Conversely, if an EV reaches a feasible energy level to complete its trip, the reward becomes less negative, reflecting successful and efficient planning. This reward formulation supports learning efficient, cost-effective, and congestion-aware charging policies within a multi-agent framework.

Discount Factor: The discount factor $\gamma \in (0, 1]$ is a key component in the MDP formulation, determining how future rewards are weighted relative to immediate ones. A value of γ close to 1 encourages agents to consider long-term consequences of their actions, while a value closer to 0 places more emphasis on immediate rewards.

In the context of the EV charging recommendation system, the discount factor enables each EV agent to (i) anticipate and plan for downstream effects of charging decisions, such as station congestion or travel delays; (ii) make strategic choices that minimize cumulative costs across multiple time steps, rather than focusing solely on short-term gains; and (iii) learn policies that are robust to delayed outcomes, such as future traffic conditions or availability at alternate stations.

This MDP formulation provides a structured foundation for implementing MARL, where agents continuously interact with a dynamic environment that incorporates real-time traffic and charging station data. The inclusion of a

discount factor ensures that agents optimize not only for immediate benefits (e.g., proximity of a station), but also for longer-term goals such as reaching the destination efficiently while avoiding overused or costly stations.

4.2. Environment Design for MARL

Environment Design and Episode Structure: We design the EV charging environment to reflect the dynamic, uncertain, and congested nature of real-world charging coordination problems. The environment simulates a typical day starting from an initial time (e.g., early morning) and progresses in discrete time steps of fixed duration (e.g., 15 min), which are aligned with realistic charging and travel intervals. An episode terminates when all EVs either reach their destination with sufficient charge or the time horizon is reached.

At initialization, the environment loads key variables, including the origin, destination, and initial SOC of each EV; energy consumption rate and battery capacity of each EV; charging power, cost, time-equivalent value, and unit availability at each station; travel distances and estimated times between EVs, stations, and destinations (computed using Euclidean distance and average travel speed).

The environment also integrates dynamic events, such as road accidents, which are simulated at predefined random locations and times. These incidents increase the travel time of affected EVs by altering the corresponding travel distance-to-time mappings. When an EV is on a route that intersects with an accident zone, its expected arrival time at a station or destination increases. As a result, the previous plans may be suboptimal and the environment would encourage rerouting to alternative stations that reduce overall cost and delay by reoptimizing the problem. These accident events are generated through a scenario generator that can be driven by external traffic data or stochastic models, allowing realistic modeling of incident occurrences. When an accident is triggered, the affected road segment experiences reduced effective speed, which is immediately reflected in the state information observed by the agents. The impact of accidents is not modeled through a separate reward term; instead, it is implicitly captured through increased travel time costs and potential congestion at alternative charging stations, thereby influencing the overall reward signal and encouraging adaptive re-planning.

At each time step, EV agents act simultaneously and select one of the following two actions: Skip charging (no-op) and continue progressing toward the destination; Choose a reachable charging station and receive energy within the limits of the station's power and their battery capacities.

The environment maintains a station load matrix that tracks usage per time step and enforces a penalty if the number of EVs at a station exceeds its charging unit capacity. Charging only begins once the EV has arrived at the station, and charging time is calculated based on the amount of energy transferred and the station's power.

The reward function for each EV is based on a weighted combination of: Travel time to the station and from the station to the destination; Charging time and wait time at the station (based on station load); Monetary charging cost and its time-equivalent value at the selected station; A small penalty for each time step to favor early and efficient decisions; A queue overload penalty when station usage exceeds capacity.

If an EV gains enough charge to complete its trip, it exits the system and receives terminal feedback. EVs that fail to do so by the end of the episode receive a large penalty. The episode terminates when all EVs reach a feasible end state or the global time horizon is exhausted. This design provides a high-fidelity testing ground for MARL to learn efficient, scalable, and adaptive charging coordination strategies under real-world conditions, including congestion, limited infrastructure, and unexpected disruptions like traffic accidents.

To facilitate the training and evaluation of the MAPPO-based EV charging coordination strategy, we construct a discrete-time, multi-agent simulation environment. The environment adheres to the PettingZoo parallel API specification [21], enabling simultaneous decision-making across agents and supporting centralized-training decentralized-execution (CTDE).

Environment Configuration: The simulation environment models a fleet of I electric vehicles and J charging stations over a finite horizon of T discrete time steps. Each time step represents a fixed duration of 15 minutes. Each EV i is characterized by agent-specific parameters, including initial SOC SOC_i, battery capacity E_i^{cap}, minimum required SOC at destination $E_i^{\min}$, and energy consumption rate per kilometer E_i. Each charging station j is defined by charging power P_j, per-unit energy cost R_j, time-equivalent cost TEC_j, and number of charging units CU_j. These parameters are initialized from empirically realistic values or prior optimization benchmarks and remain fixed throughout each episode.

Agent Dynamics and Decision Process: At each time step, agents select an optimal action from a finite action space: selecting a charging station from the available set, or continuing transit without initiating charging. Agent transitions are governed by a piecewise deterministic process based on spatial-temporal variables. If an agent chooses to travel to a charging station, it incurs a delay determined by the travel time matrix T_{ij}. Upon arrival, charging duration is computed based on energy needs and available station power, constrained by both vehicle and

station limits. Each agent is permitted to charge at most once per episode. Energy dynamics are modeled as follows: Upon travel, SOC is depleted by distance multiplied by the consumption rate; Upon charging, SOC increases by $P_j \times \Delta t$, bounded by capacity constraints.

Assignments are further constrained by reachability and temporal feasibility: an EV may only be assigned to a charging station if it can physically arrive with sufficient SOC, and the assignment must occur at or after the discretized arrival time.

Queuing Constraints and Penalties: Charging stations are subject to capacity limits, and excess assignments beyond the available number of chargers CU_j result in a queue penalty. Moreover, to promote temporally efficient behavior, a small delay penalty $\epsilon \cdot t$ is included in the reward function to encourage early-stage decisions and align agent objectives with system-level efficiency.

Perturbation-Based Real-Time Dynamics: To emulate the dynamics and uncertainties of the traffic network, we introduce a stochastic perturbation mechanism to the travel time matrices. At the start of each episode, each travel time value is reduced by a randomly sampled percentage from the interval $[0\%, 10\%]$. This mimics real-time navigation improvements, consistent with rerouting behavior observed in systems such as Google Maps.

5. MAPPO Algorithm Implementation

To address the EV charging coordination problem under dynamic traffic and system constraints, we implemented MAPPO using the open-source repository `on-policy` developed by Yu et al. [22]. This section details the integration of MAPPO with our custom `EVChargingEnv`, including environment configuration, policy adaptation, constraint enforcement, and the training-evaluation loop.

The MAPPO implementation is built upon a PettingZoo-compatible parallel environment, `EVChargingEnv`, where each EV is modeled as an autonomous learning agent. We wrap this environment using a custom `WrappedEVChargingEnv` class to define a shared observation space compatible with MAPPO's centralized critic. The environment is vectorized using `DummyVecEnv` with a single rollout thread to simplify debugging and ensure reproducibility.

Each agent receives a local observation represented as a flattened vector consisting of: (i) binary reachability indicators for each station based on the agent's SOC; (ii) normalized arrival time estimates to each station; (iii) per-station usage ratios; (iv) station-specific cost parameters; (v) agent-specific SOC; (vi) binary variable indicating if EV gets charged; and (vii) binary variable for accident on the road. The action space is discrete and contains $J + 1$ actions, in which the first action denotes a no-op (i.e., continuing without charging) and the remaining J actions describe charging at individual stations.

To support MAPPO training, we implemented custom `step` and `reset` functions that transform batched policy outputs into per-agent action dictionaries and format the resulting observations, rewards, and terminal signals into the expected tensor structures. These functions also handle agent-specific logic, such as identifying whether an EV is eligible to charge or should terminate based on its SOC and travel feasibility. We provided the list of hyperparameters that we used in Table 2.

Training configuration parameters include a total of 12,000 environment steps, 12-step episode lengths, entropy regularization (coefficient 0.2), and on-device training using GPUs when available. Logging is performed via Weights & Biases [23], capturing reward curves and policy behavior metrics.

One of the core challenges in multi-agent EV coordination is feasibility enforcement. We implement a domain-specific action masking strategy to handle constraints. Specifically, agents that have already been charged are restricted to the no-op action; Agents required to charge due to low SOCs are prohibited from selecting no-op until charging is feasible; Stations that are unreachable due to insufficient SOCs of EVs are masked out. These masks are passed to the MAPPO actor for constrained action sampling and are also updated dynamically throughout the episode.

To enable compatibility with the MAPPO framework, we developed a customized `EVRunner` class, which extends the standard `MPERunner` in the on-policy repository. This runner manages trajectory collection, computes returns using Generalized Advantage Estimation (GAE), and performs PPO-style updates. It also integrates reward shaping logic aligned with the mathematical model's objective function, including penalties for travel time, waiting time, charging time, energy cost, and station overloads.

During evaluation, the trained actor and critic models are loaded to simulate deterministic policy rollouts. We extract key metrics such as per-agent rewards, charging decisions, and SOC evolution over time. Visualization utilities generate line plots of SOC levels, station selection heatmaps, and charging logs. These tools help evaluate policy behavior and confirm alignment with theoretical expectations.

This MAPPO integration bridges a realistic, dynamic EV environment with a high-performing cooperative

MARL framework. It provides a scalable basis for benchmarking learned behaviors against static optimization models such as Gurobi under real-time traffic perturbations.

Table 2. Training hyperparameters for MARL agents.

Hyperparameter	Value
Algorithm	MAPPO
Total environment steps	24,000
Discount factor (γ)	0.99
GAE parameter (λ)	0.95
Learning rate (actor)	5×10^{-4}
Learning rate (critic)	5×10^{-4}
PPO epochs	15
Clip parameter	0.05
Entropy coefficient	0.2
Value loss coefficient	1.0
Hidden units per layer	64
Number of hidden layers	1
Nonlinearity	ReLU
Use recurrent policy	True
Number of recurrent layers	1
Data chunk length	10
Max gradient norm	10.0
Optimizer epsilon	1×10^{-5}
Weight decay	0

Note: All hyperparameters listed in this table are taken directly from the experimental configuration used in our MAPPO implementation and are shared across all agents under the CTDE framework. The values were selected based on standard practice in MARL literature and preliminary tuning to ensure stable training and convergence.

6. Experimental Results and Analysis

6.1. Environment Configuration

To evaluate the performance of our MAPPO-based EV charging coordination framework, we constructed a realistic, discrete-time multi-agent simulation environment modeled after a typical urban road network. The environment simulates real-world complexities such as spatial distances, traffic dynamics, station congestion, and agent-specific battery constraints. Each episode represents a full planning horizon segmented into discrete time steps, during which all agents act simultaneously. All training experiments were conducted on a Windows 11 system equipped with an NVIDIA GeForce RTX 4060 GPU (8.6 GB VRAM, 3072 CUDA cores, Ada architecture) and 32 GB of RAM. The machine utilized a 13th Gen Intel Core i7-13700 processor with 16 cores and 24 threads. The Python environment was configured using Python 3.12.7 within an Anaconda environment, and training scripts were executed using CUDA version 12.7.

The key parameters and configurations of the environment include: number of EVs (I), number of charging stations (J), episode Length (T) with each discrete time step representing 15 minutes ($\Delta t = 0.25$ hours), EV parameters (i.e., initial state of charge SOC_i, normalized battery capacity $E_i^{\text{cap}} = 1.0$, minimum required SOC at destination $E_i^{\min} = 0.2$, energy consumption rates vary per vehicle $E_i \in [0.04, 0.07]$ KWh/mile), and charging station parameters (i.e., charging power $P_j \in [7, 25]$ kW/hour per station, charging cost R_j ranging from 0.3 to 0.4 \$/kWh, time-equivalent cost TEC_j ranging from 0.02 to 0.04, and station capacity CU_j is based on available charging units per station), geospatial coordinates (i.e., each EV is initialized with an origin and a unique destination, station and vehicle positions are defined by latitude/longitude pairs, and distances are calculated using the Euclidean distance formula), dynamic travel conditions (i.e., travel times are derived from geodesic distances assuming a constant average speed), and system weights and penalties (i.e., time penalty coefficient $\alpha_1 = 0.6$, cost penalty coefficient $\alpha_2 = 0.4$, small delay penalty $\epsilon = 5 \cdot t$ and queue penalty of 10.0 applied when station capacity is exceeded). The parameters α_1 and α_2 are selected to balance user-centric objectives, reflecting the trade-off between travel efficiency and charging cost. The chosen values were determined through preliminary experiments to ensure stable convergence and to avoid dominance of any single reward component. We further observe that moderate variations in these parameters do not significantly affect the overall learning behavior, indicating that the proposed framework is reasonably robust to their selection. The initial state of charge SOC_i is normalized and defined within the range $[0.4, 1]$, indicating that each EV begins with at least 40% and at most 100% of its full battery capacity.

To simulate real-world disruptions, we incorporate dynamic accident scenarios that affect EV routing. Each episode assumes that an accident occurs along at least one EV's route from its origin to the assigned charging station. Specifically, the accident is positioned randomly between 50% and 70% of the total travel distance from the origin to the station, representing a mid-route disruption. At the time of the accident, each EV traveling toward a station has a randomly assigned probability (between 30% and 40%) of being notified of the incident. This models the behavior of real-time navigation systems, such as Google Maps, which notify drivers of traffic incidents and suggest alternative routes. Notified EVs can re-evaluate their charging assignment and potentially reroute to a more efficient station, while others continue along their original path, possibly incurring increased travel time due to the disruption. This configuration enables a realistic simulation of both individual EV behavior and overall system-level congestion. The modular design of the environment supports easy expansion to larger fleet sizes, additional stations, and varied topologies. Figure 3 visualizes the EV routes, charging stations, destinations, and the accident location.

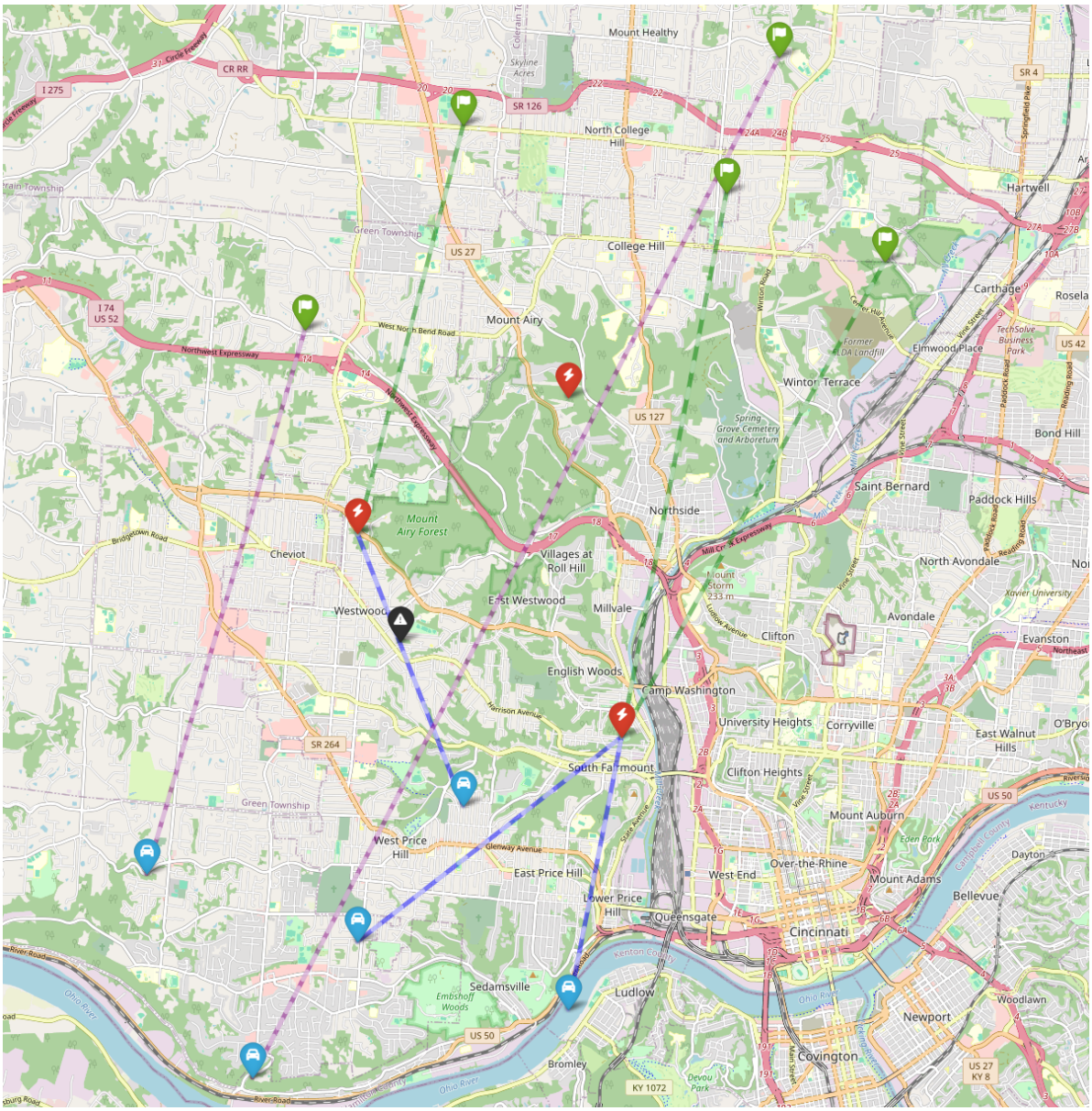

Figure 3. Visualization of EV routes, charging stations, destinations, and the accident location. EVs either travel directly to their destination or stop at a charging station. One route is affected by an accident, requiring potential rerouting.

6.2. Performance Comparison with Gurobi Baseline

To evaluate the effectiveness of our MARL-based approach, we compare it against a static optimization benchmark implemented in Gurobi. The Gurobi model uses the same mathematical formulation described in Section III but assumes deterministic travel times without real-time perturbations or accident disruptions, with its details provided in the Appendix. In contrast, MAPPO agents interact with a dynamic environment where travel times are perturbed by accidents that can delay routes. Key performance metrics include:

- **Total cost per episode:** Combined travel, waiting, charging, and delay costs.
- **Average travel time:** Time from origin to destination, including charging detours.
- **Queue overload occurrences:** Frequency of station congestion events.
- **Success rate:** Percentage of EVs that reach their destination with sufficient SOC.

6.3. Case Study: Accident-Triggered Reassignment

This case study considers a scenario involving five EVs and three charging stations. Based on the initial SOC levels, not all EVs require charging. Specifically, only EV_1 and EV_2 are unable to reach their respective destinations with the minimum required SOC, while the other three EVs can travel directly from their origin to their destination without recharging.

We first evaluated the system using our exact optimization model implemented in Gurobi. The optimal assignment, in this case, designates EV_1 to Station 1 and EV_2 to Station 2.

Subsequently, we solved the same configuration using our proposed MARL model. The resulting assignments and trajectories, visualized in Figure 4, reflect the decisions made by the MARL policy. The total cost in the MARL approach is 0.18 for Agent 1 and 0.42 for Agent 2, resulting in a combined cost of 0.6, which matches the total cost obtained from the exact optimization approach.

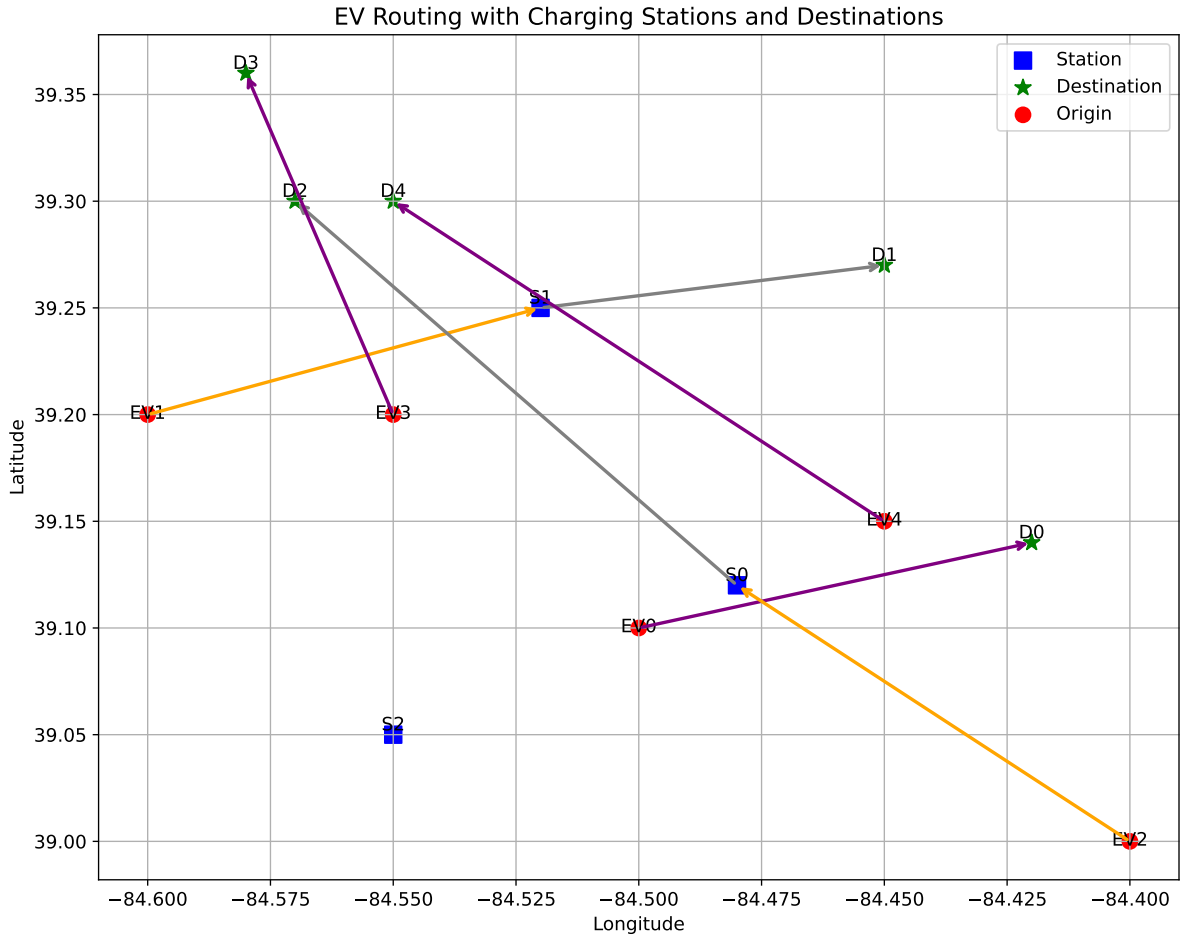

Figure 4. Graphical representation of route assignment for EVs without accident.

As the EVs begin traveling along their assigned route either to a charging station or directly to their destinations, an unexpected traffic incident occurs within the city. We assume that this accident intersects the route of at least one EV. Upon receiving updated traffic information (e.g., via Google Maps), the affected EVs are notified, and the model updates the positions and SOC levels for all EVs. Figure 5 shows the updated position of EVs after starting their route and being notified of an accident on the route of one of the vehicles.

Due to the increased travel time resulting from the accident, the model re-evaluates the system and resolves the optimization problem against the new configuration. It may determine that staying on the original route is still optimal despite the delay, or it may recommend rerouting to a different station if doing so yields a more cost-effective result. If the travel time after the accident becomes more than twice the expected time—calculated by dividing the updated distance by speed—the model reassigns EV_2 to a different station with improved cost

efficiency. As we can see in Figure 6, the decisions are updated for all the vehicles after the accident happens.

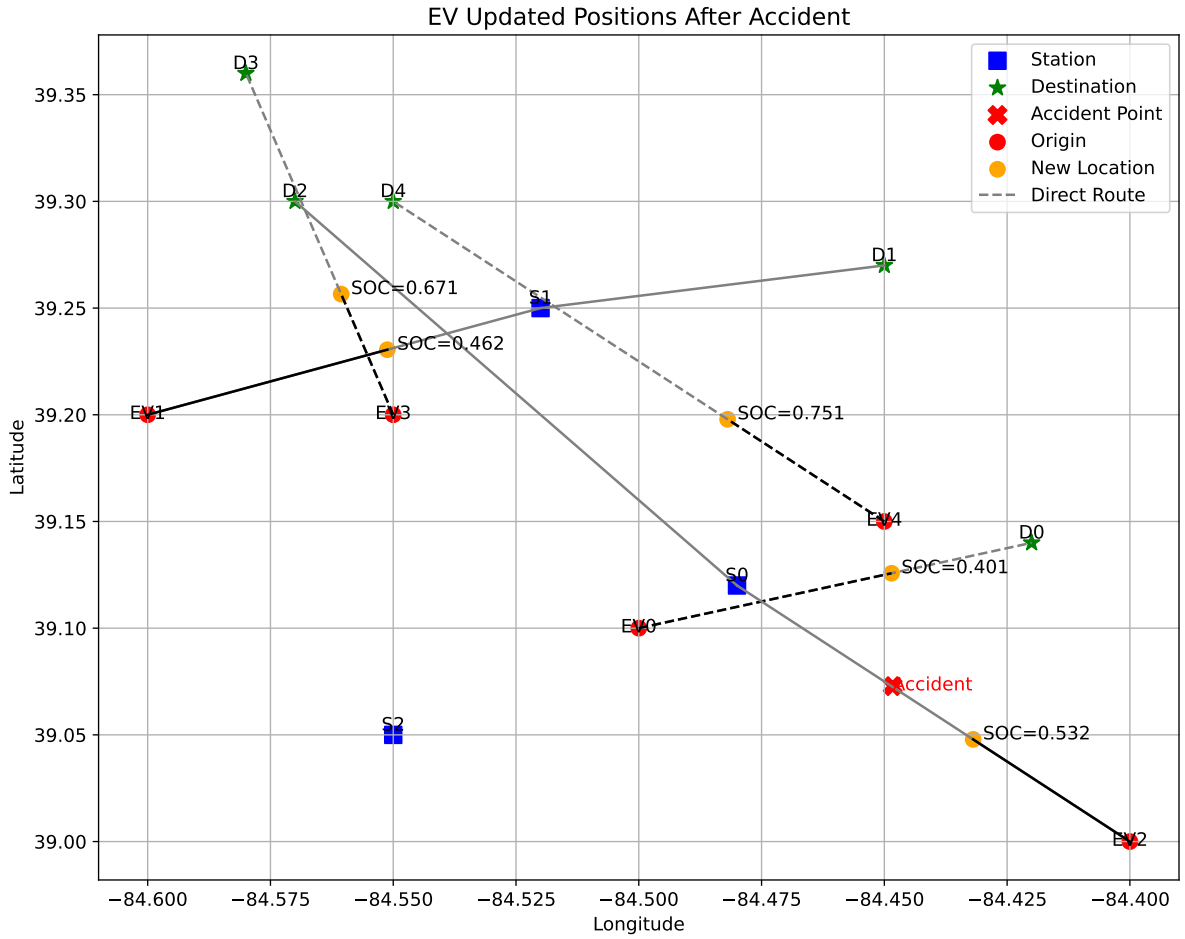

Figure 5. Updated positions and SOC levels of EVs after the accident.

Simulation results show that the MAPPO-trained agents outperform the Gurobi baseline under dynamic scenarios. While Gurobi provides globally optimal plans under static settings, its performance degrades in actual situations with the presence of accidents and/or stochastic traffic updates. MAPPO demonstrates higher adaptability by rerouting affected EVs to alternate stations and avoiding congested locations, thus offering reduced cost and improved user satisfaction. The impact of accident events and the effectiveness of MARL-based reassignment are summarized as follows:

- **Average additional delay per vehicle:** In the absence of MARL-based reassignment, vehicles affected by an accident experience a noticeably higher average delay due to increased travel time and congestion at charging stations. With MARL coordination, agents adapt their charging decisions based on updated traffic conditions, resulting in a significantly reduced average delay per vehicle.
- **Queue formation at charging stations:** Without reassignment, a large number of EVs converge on the originally selected charging station, leading to congestion and long queues. Under the MARL framework, vehicles are redistributed across alternative stations, reducing the number of vehicles experiencing queueing and shortening overall waiting times.
- **Total system cost:** We evaluate the total system cost, defined by the cumulative reward components including travel time, charging time, and queue-related penalties. Results show that system cost increases sharply after an accident when reassignment is not allowed, whereas the MARL-based approach effectively mitigates this increase by dynamically adjusting vehicle-to-station assignments.
- **Adaptive response to disruptions:** These observations demonstrate that the proposed MARL framework can respond effectively to unexpected traffic disruptions by re-optimizing agent decisions online, improving robustness compared to non-coordinated or static charging strategies.

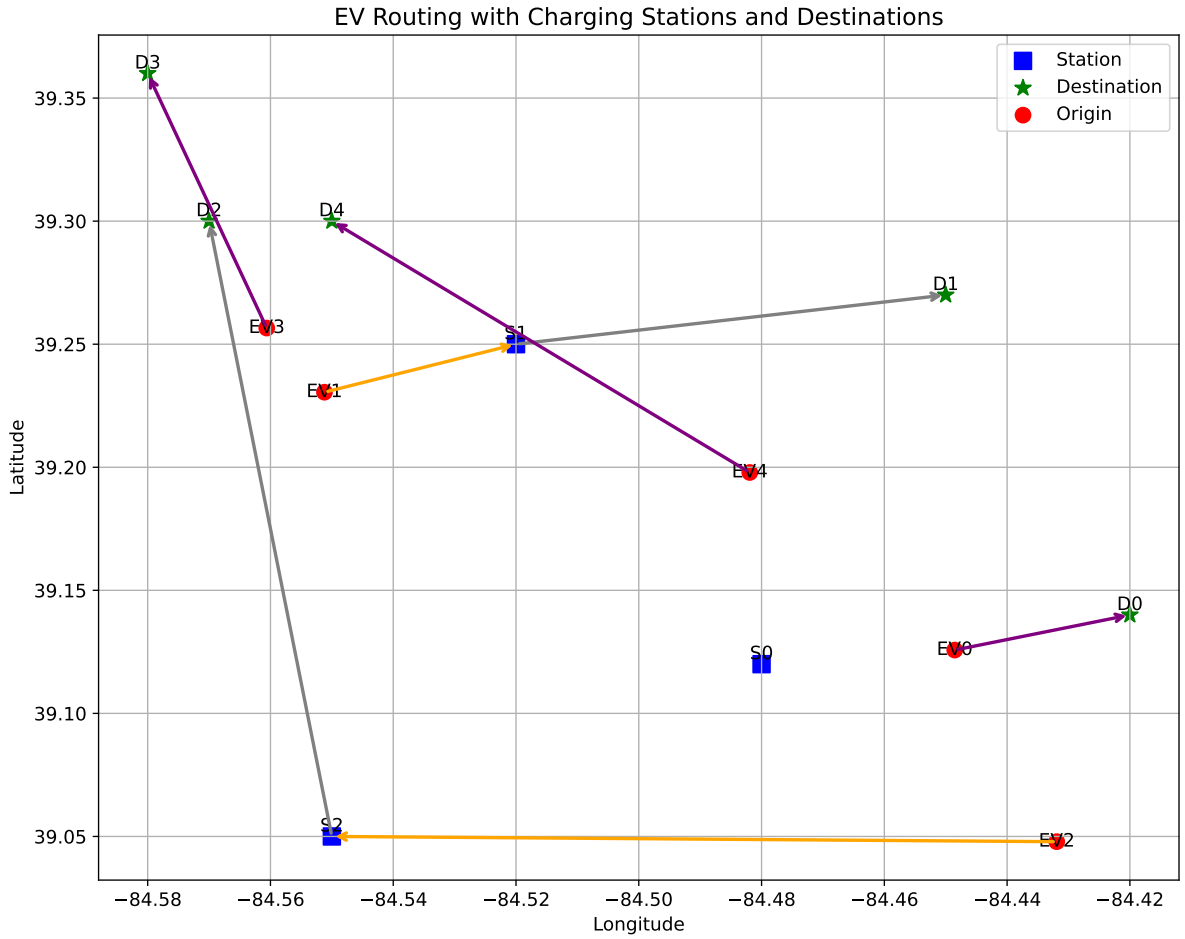

Figure 6. Graphical representation of route assignments after the accident.

6.4. Experiments Discussion

While the numerical case study uses a modest fleet and a small set of charging stations, it is intended as a controlled proof-of-concept to validate the Markov game formulation, feasibility enforcement (via action masking), and the ability of MAPPO to reproduce optimal behavior in static settings and adapt under dynamic disruptions (e.g., accident-triggered travel-time changes). Importantly, the proposed CTDE-based MARL framework is designed for scalable deployment: after offline training, execution requires only per-agent neural-network inference (a forward pass) rather than repeated mixed-integer re-optimization, which becomes increasingly expensive as the number of EVs and stations grows. Moreover, the effective action set is typically much smaller than the total number of stations due to reachability- and constraint-based masking at each decision step. Large-scale benchmarking on city-level networks is therefore a natural next step and is left for future work.

Figure 7 presents the exponential moving average (EMA) of the critic gradient norm over training steps for the scenario involving 50 electric vehicles (EVs) and 20 charging stations. A smoothing factor of 0.8 was applied to reduce noise and reveal the underlying trend. Initially, the critic gradient norm is high, indicating substantial parameter updates during early training. Over time, the gradient norm steadily decreases, suggesting convergence and improved training stability. This trend demonstrates that the critic network is learning effectively.

In Figure 8, we present the exponential moving average (EMA) of the average episode rewards obtained throughout the training process. The smoothing helps to highlight long-term trends by reducing short-term fluctuations in the reward signal. The plot shows a generally increasing trend in average reward over time, which is indicative of learning progress and policy improvement. Despite some initial variability—typical in multi-agent reinforcement learning—the curve stabilizes in later stages, suggesting convergence toward a more optimal and consistent joint policy. This behavior indicates the agents' growing ability to coordinate effectively under the reward structure defined by the environment.

To evaluate the benefit of the proposed MARL framework, we compare the total system cost under two settings: (i) a baseline scenario in which no rerouting is performed after an accident happens; and (ii) the proposed approach where agents adapt their decisions using the learned MARL policy when an accident happens. We repeat this

experiment for 10 independent runs. The results show that, on average, the MARL-based approach decreases the total system cost by approximately 21% compared to the no-rerouting baseline. This improvement highlights the ability of the learned policy to effectively respond to dynamic disruptions by rerouting affected EVs to alternate stations. Overall, these findings show that the MARL framework not only learns efficient charging strategies, but also provides significant performance gains in the presence of real-time disruptions such as traffic incidents.

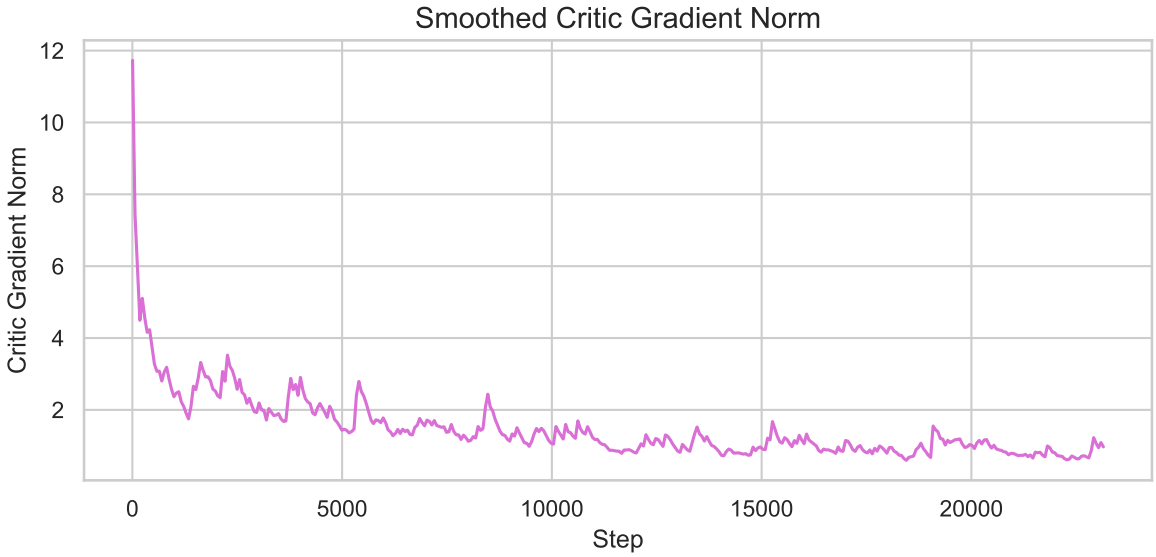

Figure 7. Critic gradient norm during training. The exponential moving average is used for smoothing.

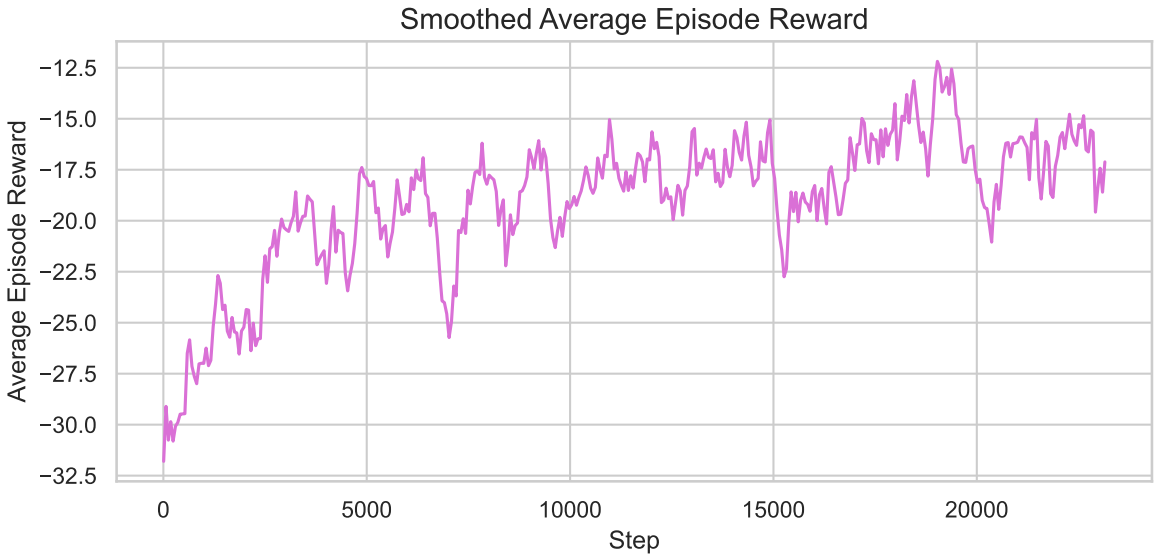

Figure 8. Smoothed average episode reward over training steps. The curve reflects the learning progress of the agents, showing an overall increasing trend that stabilizes over time, indicating convergence and improved policy performance.

7. Conclusions

This paper proposes an adaptive and robust EV charging coordination strategy based on MARL, specifically using the MAPPO framework. By integrating dynamic traffic conditions, charging station characteristics, and EV-specific constraints into the decision-making process, the proposed approach enables decentralized, real-time, and scalable charging strategies. The simulation results demonstrate that our MARL model outperforms traditional static optimization methods, such as Gurobi, particularly in scenarios involving real-time disruptions like accidents. The environment design and training framework support adaptive behaviors that reroute EVs dynamically to reduce congestion, waiting time, and energy costs. This work bridges the gap between theoretical optimization and practical deployment of intelligent EV charging strategies under uncertainty. Future research can extend this framework by incorporating more complex traffic prediction models, multi-objective optimization, vehicle-to-grid interactions, and generalization to larger-scale transportation networks.

Author Contributions

S.R.: writing—original draft preparation, investigation, validation, formal analysis, methodology, software, data curation, conceptualization; H.W.: writing—review & editing, writing—original draft preparation, validation, supervision, resources, investigation, methodology, formal analysis; L.W.: writing—review & editing, writing—original draft preparation, validation, supervision, resources, investigation, methodology, formal analysis. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

Not applicable.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

During the preparation of this work, the authors used ChatGPT-5 to polish writing and fix typos. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Appendix A. Gurobi-Based Optimization Benchmark

To establish a static performance baseline, we implement the EV charging optimization problem as a mixed-integer programming (MIP) problem and solve it via Gurobi. The optimization problem captures key temporal and spatial constraints, as well as system-level objectives, based on the mathematical model defined in Section 3.

The objective function minimizes the weighted sum of travel time, charging time, and energy cost, subject to prevailing constraints described in Section 3. Specifically, the energy reachability constraints ensure that EVs can only be assigned to reachable stations based on current SOC levels and locations; the charging feasibility constraints guarantee that the energy charged must be sufficient to reach the destination with the required SOC; the timing logic constraints indicate that EVs may only begin charging after arriving at the station; the station capacity constraints describe that at any time step, the number of assigned EVs must not exceed the available units; the overload penalties present that excess assignments are penalized through an auxiliary term in the objective. Binary variables x_{ijt} are used to represent the assignment of EV i to station j at time t, while continuous variables govern the amount of energy charged (E_{ijt}^{charge}), charging duration (T_{ijt}^{charge}), and station overload levels.

This deterministic MIP model is solved using a static travel time matrix, assuming all inputs (e.g., distances, costs, and initial SOCs) for the entire day remain fixed and known in advance. This allows Gurobi to compute a globally optimal solution under deterministic conditions. However, this deterministic MIP model does not adapt to real-time perturbations or dynamic traffic conditions. It assumes full observability and static parameters, which limit its applicability in dynamic or uncertain environments. In contrast, the MAPPO model is trained directly in a stochastic setting, with perturbations to travel time incorporated during training episodes. This enables MAPPO to learn adaptive strategies that respond to varying conditions and potentially outperform the deterministic MIP model in scenarios where the initial plan becomes suboptimal due to external changes. In our experiments, we evaluate the deterministic MIP solution on the same perturbed environment used for MARL evaluation, allowing a fair and consistent comparison of static versus adaptive decision-making frameworks.

References

1. IEA. Global EV Outlook 2020. Available online: https://www.iea.org/reports/global-ev-outlook-2020 (accessed on 15 June 2020).

2. Suanpang, P.; Jamjuntr, P. Optimizing Electric Vehicle Charging Recommendation in Smart Cities: A Multi-Agent Reinforcement Learning Approach. *World Electr. Veh. J.* **2024**, *15*, 67.
3. Park, K.; Moon, I. Multi-Agent Deep Reinforcement Learning Approach for EV Charging Scheduling in a Smart Grid. *Appl. Energy* **2022**, *328*, 120111.
4. Su, S.; Li, Y.; Yamashita, K.; et al. Electric Vehicle Charging Guidance Strategy Considering "Traffic Network-Charging Station-Driver" Modeling: A Multiagent Deep Reinforcement Learning-Based Approach. *IEEE Trans. Transp. Electrif.* **2023**, *10*, 4653–4666.
5. Buşoniu, L.; Babuška, R.; De Schutter, B. Multi-Agent Reinforcement Learning: An Overview. *Innov. Multi-Agent Syst. Appl.* **2010**, 183–221.
6. Kraemer, L.; Banerjee, B. Multi-Agent Reinforcement Learning as a Rehearsal for Decentralized Planning. *Neurocomputing* **2016**, *190*, 82–94.
7. Busoniu, L.; Babuska, R.; De Schutter, B. A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2008**, *38*, 156–172.
8. Lu, C.; Bao, Q.; Xia, S.; et al. Centralized Reinforcement Learning for Multi-Agent Cooperative Environments. *Evol. Intell.* **2024**, *17*, 267–273.
9. Lowe, R.; Wu, Y.I.; Tamar, A.; et al. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
10. Foerster, J.; Farquhar, G.; Afouras, T.; et al. Counterfactual Multi-Agent Policy Gradients. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
11. Wen, M.; Kuba, J.; Lin, R.; et al. Multi-Agent Reinforcement Learning Is a Sequence Modeling Problem. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 16509–16521.
12. Zhang, W.; Liu, H.; Xiong, H.; et al. RLCharge: Imitative Multi-Agent Spatiotemporal Reinforcement Learning for Electric Vehicle Charging Station Recommendation. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 6290–6304.
13. Li, Y.; Su, S.; Zhang, M.; et al. Multi-Agent Graph Reinforcement Learning Method for Electric Vehicle On-Route Charging Guidance in Coupled Transportation Electrification. *IEEE Trans. Sustain. Energy* **2024**, *15*, 1180–1193.
14. Zhang, Z.; Wan, Y.; Qin, J.; et al. A Deep RL-Based Algorithm for Coordinated Charging of Electric Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 18774–18784.
15. Suanpang, P.; Jamjuntr, P.; Jermsittiparsert, K.; et al. Adaptive Multi-Agent Reinforcement Learning for Optimizing Dynamic Electric Vehicle Charging Networks in Thailand. *World Electr. Veh. J.* **2024**, *15*, 453.
16. Maria, E.; Budiman, E.; Taruk, M.; et al. Measure Distance Locating Nearest Public Facilities Using Haversine and Euclidean Methods. In Proceedings of the International Conference on Applied Science and Technology (iCAST on Engineering Science), Bali, Indonesia, 24–25 October 2019; Volume 1450, p. 012080.
17. Sutton, R.S.; Precup, D.; Singh, S. Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. *Artif. Intell.* **1999**, *112*, 181–211.
18. Bacon, P.L.; Harb, J.; Precup, D. The Option-Critic Architecture. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31, pp. 1726–1734.
19. Dulac-Arnold, G.; Evans, R.; van Hasselt, H.; et al. Deep Reinforcement Learning in Large Discrete Action Spaces. *arXiv* **2015**, arXiv:1512.07679.
20. Hausknecht, M.; Stone, P. Deep Reinforcement Learning in Parameterized Action Space. In Proceedings of the International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2–4 May 2016.
21. PettingZoo Documentation. Available online: https://pettingzoo.farama.org/index.html (accessed on 31 August 2025).
22. Yu, C.; Velu, A.; Vinitsky, E.; et al. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24611–24624.
23. Weights & Biases AI developer platform. Available online: https://wandb.ai/ (accessed on 31 August 2025).