A Deep Q-Network Framework for Joint Optimization of EV Charging Station Placement and Vehicle Routing

Iakovos Ioannou*, Christophoros Christophorou†, Christina (Tanya) Politi§, Spyros Denazis‡,
Marios Raspopoulos†, Vasos Vassiliou*¶
* Department of Computer Science, University of Cyprus and CYENS - Centre of Excellence, Cyprus
† University of Central Lancashire (UCLAN), Larnaca, Cyprus
‡ University of Patras, Greece
§ University of Peloponnese, Greece

***Abstract*—The rapid adoption of electric vehicles (EVs) presents new challenges in designing efficient charging infrastructure and route planning mechanisms to ensure sustainable urban mobility. This paper introduces a novel two-stage optimization framework that leverages Deep Q-Networks (DQNs) for the intelligent placement of EV charging stations and real-time routing of vehicles. In the first stage, a DQN placement model learns to identify optimal charging station locations within a graph-represented road network, minimizing the expected energy consumption of future routing. In the second stage, a separate DQN routing model is trained to approximate cost-to-go functions and guide vehicles to the nearest stations efficiently. The models are evaluated against traditional methods, including Q-Learning, neural networks (NN), deep graph neural networks (DGNN), and Random Walk baselines. Simulation results across diverse network topologies demonstrate that our DQN approach consistently outperforms baselines in terms of average energy consumption, travel time, and route success ratio. Specifically, the DQN placement strategy achieved the lowest average energy consumption of 1.2387 kWh and the most spatially equitable configuration (average distance: 6.34 km), while the DQN routing model recorded the best performance with 1.2587 kWh average energy, 1.98 hops, and a 96.67% success rate. These findings highlight the effectiveness of deep reinforcement learning in enhancing the scalability, reliability, and energy efficiency of EV infrastructure planning.**

***Index Terms*—Electric Vehicles (EV), Deep Q-Network (DQN), Charging Station Placement, Route Optimization, Reinforcement Learning, Smart Mobility, Energy Efficiency**

I. Introduction

The accelerating global adoption of electric vehicles (EVs) is driving a transformative shift in sustainable urban transportation [1]. However, this transition introduces formidable challenges in optimizing limited charging infrastructure, especially in complex and dynamic urban environments [2]. High-density cityscapes exhibit erratic EV traffic patterns and spatially diverse charging demands [3], often resulting in long queues, station congestion, and critical energy-depletion events. These issues erode user trust and inhibit the scalability of EV ecosystems [4]. This work targets the dual optimization challenge central to EV-based smart mobility: (i) the strategic placement of a fixed number of charging stations ($K = 3$ in our experimental setup) within an urban road network, represented as a directed graph $G = (V, E)$, where nodes represent candidate charging locations and edges encode travel and energy costs; and (ii) the real-time routing of EVs to these stations to minimize energy consumption, travel time, and distance [5]. The complexity of real-world traffic networks, marked by dynamic topology, varying state-of-charge levels, and stochastic demands, necessitates intelligent and scalable learning-based solutions [6].

Conventional approaches to this problem often rely on static heuristics or centralized control systems that fail to incorporate environmental feedback or adapt to dynamic changes in the network [7], [8]. As a result, popular charging locations tend to be overloaded while other stations remain underutilized, leading to inefficient vehicle flows and poor user experience [9]. To address these limitations, we propose a unified machine learning-driven framework that intelligently integrates the placement and routing problems and learns from optimal examples to adapt to future unseen scenarios [10]. This study introduces a comprehensive and modular two-phase solution that harnesses deep reinforcement learning and supervised machine learning for efficient EV charging system design. Phase one addresses charging station placement through various strategies, including a Deep Q-Network (DQN) [11] that learns placement actions via reward feedback based on simulated routing energy [12]. It is supplemented by placement Neural Networks (NN) [13], Graph Neural Networks (GNNs) [14], and machine learning (ML) proxy methods [15] that score placements using routing outcomes. In phase two, a separately trained DQN routing model approximates energy-optimal paths for EVs by learning value functions based on Dijkstra-generated ground truth [16]. Additional supervised routing models include Q-learning [17], [18], standard Neural Networks (NN) [19], and Deep Graph Neural Networks (DGNN) [14], each trained to generalize routing across variable network sizes and vehicle distributions [6].

The novelty of our work lies in this joint learning framework that co-evolves infrastructure placement and vehicle routing, exploiting learned knowledge across both tasks. By simulating a diverse set of graphs and loads, our method rigorously evaluates each model's ability to generalize and adapt. Moreover, our ML models are designed to be fully auto-tuned—trained via data-driven supervision to eliminate the need for manual tuning and perform effectively in unknown environments. The main contributions of this paper are as follows:

- **Integrated Two-Phase Optimization Framework:** A novel, unified framework for simultaneous EV charging

This work has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 739578, the METACITIES project under Grant Agreement No. 101087257, and the Government of the Republic of Cyprus through the Deputy Ministry of Research, Innovation and Digital Policy.

station placement and routing using learning-based models in a modular simulation pipeline.

- **Data-Driven and Auto-Tuned ML Models:** Deployment of DQN, NN, DGNN, and Q-Learning models that are trained using optimal path data from Dijkstra's algorithm and optimized through supervised learning and reinforcement feedback.
- **Machine Learning-Driven Placement Strategies:** Exploration of multiple placement approaches—including DQN placement, GNN-spectral analysis, and proxy-based scoring—linked to routing outcomes for adaptive infrastructure planning.
- **Extensive Evaluation across Topologies and Loads:** Detailed simulations across large graph sizes (100 nodes) and vehicle loads (100) to benchmark and analyze model performance under realistic constraints.
- **Seamless Integration of Learning and Planning:** Demonstration of how learned routing behaviors can serve as evaluators for placement policies, forming an efficient, scalable loop between planning and control.

The remainder of the paper is structured as follows: Section II defines the joint optimization problem for EV charging station placement and routing and details the system architecture. Section III presents the two-phase methodology, including synthetic data generation, DQN model training, and evaluation metrics. Section IV discusses the simulation results, analyzing the performance of DQN and baseline models across diverse network scenarios. Finally, Section V concludes the paper, highlighting key findings and outlining future research directions, including integration with digital twins and federated learning.

II. Problem Description and System Architecture

In modern urban and suburban environments, the rapid growth of electric vehicles (EVs) creates two intertwined challenges. First, planners must decide *where* to install a fixed number of charging stations so that drivers can reach them conveniently and with minimal energy expenditure. Second, given any chosen set of stations, individual EVs need an efficient way to navigate the road network—selecting routes that consume the least energy while avoiding unnecessary detours or delays. These two tasks—*station placement* and *vehicle routing*—are deeply connected: stations that are poorly sited force vehicles into long, energy-hungry trips, while even an optimal routing policy cannot overcome a suboptimal placement of chargers. Consequently, finding a jointly optimal solution means striking the right balance between broad geographic coverage and energy-efficient routing. This creates a complex combinatorial problem: choosing a handful of locations out of many possibilities, and simultaneously guiding EVs along near-optimal paths through a dynamic traffic network. The proposed architecture tackles these challenges head-on by combining classical graph-theoretic methods with deep reinforcement learning in a two-phase pipeline. First, we identify near-optimal charging-station locations; then, we learn energy-efficient routing policies. Formally, we solve the coupled two-level optimization:

- *Placement:* choose $S \subset V$, $|S| = K$, to minimize $\frac{1}{|V \setminus S|} \sum_{v \in V \setminus S} \min_{s \in S} \mathrm{Cost}(v \to s)$.
- *Routing:* for fixed S, compute or approximate energy-optimal paths from each $v \notin S$ to its nearest $s \in S$, either exactly via Dijkstra or approximately via the learned $V(\cdot)$.

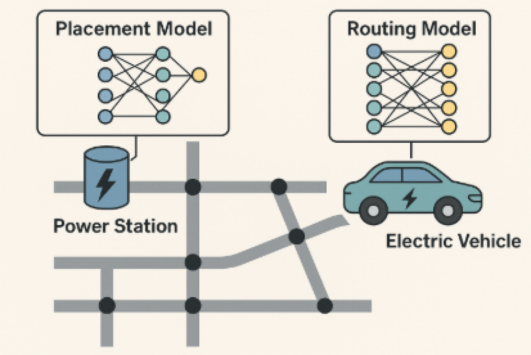

Figure 1: System Model

Figure 1 illustrates the complete workflow—from synthetic network generation through placement optimization to trained routing—over a graph-modeled road network, enabling joint station planning and vehicle navigation under a unified framework.

III. Methodology

This section outlines a comprehensive two-phase methodology for optimizing Electric Vehicle (EV) charging infrastructure, combining supervised learning and reinforcement learning techniques to improve both station placement and routing efficiency. The pipeline begins with synthetic, strongly connected road network generation and proceeds through feature extraction, model training, and performance evaluation. The learning backbone of the framework leverages DQL, which extends classical Q-learning by using deep neural networks to approximate the optimal action-value function $Q^*(s,a)$. Instead of relying on tabular Q-values, DQL generalizes over large state-action spaces through a DQN. Stability during training is ensured via experience replay (randomized mini-batch updates) and a target network (periodically updated to stabilize learning). An ε-greedy policy balances exploration and exploitation. This architecture enables DQN to make effective, real-time decisions in high-dimensional EV routing and placement tasks [11], [20], [21]. Overall, the proposed pipeline supports a scalable and generalizable approach to EV infrastructure planning, integrating simulation-based datasets, ML model training, and robust evaluation for both placement and routing tasks.

A. Dataset Generation

a) Road Network Synthesis

Synthetic urban road networks are modeled as directed graphs $G = (V,E)$, where nodes V represent intersections and edges E represent directed road segments. Each edge is annotated with distance (km_e) and energy consumption ($\mathrm{kWh}_e = 0.2 \times \mathrm{km}_e$). To ensure strong connectivity, graphs are generated from a base ring topology with additional random and targeted edges. Diverse topologies are constructed using varying node counts and random seeds to simulate different urban layouts.

b) Placement Feature Dataset

For each generated graph, a greedy Dijkstra-based heuristic is used to select an initial placement of K charging stations.

The first station is placed at the node with the minimum total distance to all others. Subsequent stations are added iteratively to minimize the overall nearest-station distance across all nodes. This process ensures spatial diversity and reduced average travel cost. For each placement configuration, a feature vector is extracted, capturing the impact of station locations on network accessibility. Specifically, for every non-station node, the shortest path to the nearest station is computed using Dijkstra's algorithm, from which several statistics are derived.Energy features include the average and standard deviation of energy required to reach the nearest station: $\text{AvgEnergy} = \frac{1}{|V\setminus S|}\sum_{v\in V\setminus S} E(v)$. Distance metrics include the average and standard deviation of the shortest path lengths: $\text{AvgDist} = \frac{1}{|V\setminus S|}\sum_{v\in V\setminus S} D(v)$. Hop count statistics (AvgHops) and time-based metrics (AvgTime) are also computed, with time estimated from distance and average speed. Together, these aggregated statistics describe the overall accessibility and energy efficiency of the placement. Additional graph-theoretic features—such as coverage within 10km and centrality metrics (e.g., degree, betweenness)—are included to enrich spatial analysis. The resulting vectors are labeled with multi-label binary targets indicating selected station nodes, forming a supervised training dataset.

c) Routing (Path-Finding) Dataset

To train routing models, a comprehensive dataset is generated using synthetically created graphs of various sizes and topologies. Each graph includes edge-level annotations for distance (km), energy cost (kWh), and estimated travel time (s), assuming average vehicle speed. Charging station placements are again derived using a greedy Dijkstra-based heuristic. For each non-station node v, the nearest station s^* is identified by minimizing the total energy cost:

$$s^* = \arg\min_{s\in S} \text{Energy}(v \to s).$$

The optimal path $\mathscr{P}_{v\to s^*}$ is extracted, and the following metrics are computed:

$$E(v) = \sum_{(u,u')\in\mathscr{P}} \text{kWh}(u,u'), \tag{1}$$

$$D(v) = \sum_{(u,u')\in\mathscr{P}} \text{km}(u,u'), \tag{2}$$

$$T(v) = \sum_{(u,u')\in\mathscr{P}} \text{time}(u,u'), \tag{3}$$

$$H(v) = |\mathscr{P}|. \tag{4}$$

Each sample is stored as a tuple $[v, s^*, E(v), D(v), T(v), H(v)]$, used to train supervised models for routing cost approximation. This dataset supports the learning of deep Q-networks (DQN) capable of generalizing routing behavior across topologies, eliminating the need for explicit shortest path computation at inference time.

B. Placement and Routing Model Training and Architecture

Multiple methods are considered for placement: greedy heuristic, spectral GNN, and NN. Although all were evaluated, the DQN ML placement method proved most effective. This work employs a Deep Q-Network-based architecture to jointly optimize the placement and routing of electric vehicle (EV) charging stations. While several methods were evaluated, including greedy heuristics and GNN-based models, the DQN-informed placement and routing approach demonstrated the best trade-off between scalability and energy efficiency.

1) Placement Model

a) DQN Placement Model Training

The DQN placement model is trained using reinforcement learning to select the optimal set of K charging station locations within a given graph. The environment state is encoded as a binary vector $\mathbf{s} \in \{0,1\}^N$, where each element indicates whether a node currently hosts a station. At each timestep, the agent selects an action $a \in \{1,\ldots,N\}$ corresponding to the next node where a station should be placed. The placement process proceeds sequentially until K stations are selected, forming a complete placement configuration.

The reward signal is derived from evaluating the total routing cost under the current placement. Specifically, for a given placement configuration, the average energy consumption for routing all non-station nodes to the nearest station is computed using the trained DQN approach. This cost is transformed into a reward (e.g., negative energy cost) to guide the agent toward more efficient placements. The Q-function $Q(s,a)$ is updated using the Bellman equation:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\left[r_t + \gamma \max_{a'} Q(s_{t+1},a') - Q(s_t,a_t)\right] \tag{5}$$

where α is the learning rate and γ is the discount factor.

The model is implemented as a multi-layer perceptron with input size N, hidden layers with ReLU activations, and an output layer of size N representing Q-values for all node actions. The training employs experience replay and ε-greedy exploration to balance exploitation and exploration. Once convergence is achieved, the agent is able to recommend near-optimal station placements for unseen graphs without needing to simulate all possible configurations. Overall, the DQN placement model was trained using 70% of the generated graph instances, while the remaining 30% were reserved for evaluation on unseen topologies to assess generalization and placement effectiveness.

b) DQN Placement Model

The strategic placement of EV charging stations is modeled as a sequential decision-making process, addressed by a Deep Q-Network (DQN) based reinforcement learning agent. This agent learns a policy to select an optimal set of K charging station locations from the available nodes V in the graph $G = (V,E)$. The core components of this RL approach are defined as follows:

State Representation. The state $s_P \in \mathscr{S}_P$ for the placement agent encapsulates information about the graph's topology and the current partial placement of charging stations. This can be represented, for instance, by a binary vector of length $|V|$ indicating which nodes have already been selected to host stations, potentially augmented with a counter for the number of stations placed so far, or by more complex graph embedding features that capture the overall network configuration.

Action Space. At each decision step t in the placement process (where t ranges from 1 to K, the total number of stations to be placed), the agent selects an action $a_P \in \mathscr{A}_P$. The action space $\mathscr{A}_P$ consists of all graph nodes V that have not yet been chosen in the current partial placement S_{t-1} (i.e., $\mathscr{A}_P = V \setminus S_{t-1}$). Executing an action a_P corresponds to selecting the chosen node to host the next charging station.

Reward Function. The reward r_P is critical for guiding the placement agent's learning process. Typically, a significant reward (or penalty) is provided only upon the completion of a full placement, i.e., after all K stations have been sited. This terminal reward reflects the overall quality of the chosen set of K station locations, $S_K = \{a_{P,1}, \dots, a_{P,K}\}$. The quality is quantified by the efficiency of routing vehicles to these stations. Specifically, the reward can be defined as the negative of the average routing energy (or a similar metric like travel time or unreachability penalty) calculated for a representative set of vehicles or all non-station nodes when routed to their nearest station in S_K. These routing costs can be estimated using a pre-trained DQN routing model, albeit computationally intensive, ground-truth evaluation during RL training. The reward is formulated as: $r_P = -\frac{1}{|V \setminus S_K|} \sum_{v \in V \setminus S_K} \text{EstimatedRoutingCost}(v \to S_K)$.

A deep neural network is utilized to approximate the optimal action-value function $Q_P^*(s_P, a_P)$, representing the maximum expected cumulative reward for taking action a_P in state s_P and following the optimal policy thereafter. The network maps states s_P to Q-values for all possible actions. It is trained by minimizing the temporal difference error, with updates guided by the Bellman equation:

$$Q_P(s_{P,t}, a_{P,t}) \leftarrow Q_P(s_{P,t}, a_{P,t}) \tag{6}$$

$$+ \alpha \Big[r_{P,t} + \gamma \max_{a'} Q_P(s_{P,t+1}, a') - Q_P(s_{P,t}, a_{P,t}) \Big] \tag{7}$$

where α is the learning rate and γ is the discount factor. Standard RL techniques such as experience replay (storing and sampling transitions (s_P, a_P, r_P, s'_P)) and an ε-greedy exploration strategy are employed during training, which proceeds over numerous episodes, each involving the placement of K stations.

2) Routing Model

While our placement model relies on Q-learning with discrete actions $Q(s,a)$, the routing model instead estimates a scalar value function $V(s)$, as the objective is to evaluate expected energy cost from a node to its nearest station rather than choosing between multiple actions.

a) DQN Routing Model Training

The routing model is trained as a value-based deep reinforcement learning model that approximates the state-value function $V(s)$, where each state s represents a node in the graph and the value $V(s)$ denotes the expected minimum cumulative energy required to reach the nearest charging station. The input is a one-hot encoded vector $\mathbf{s} \in \mathbb{R}^N$ representing the current node. The output is a vector $\mathbf{v} = V(s')$, where each component v_i estimates the minimum cumulative energy required to reach the nearest charging station from node i.

Training data is generated using Dijkstra's algorithm to compute optimal energy paths from each non-station node to its nearest station. Let v_i^* be the target cost from node i. The DQN is trained to minimize the mean squared error (MSE) between the predicted and actual values:

$$\mathscr{L}_{\text{routing}} = \frac{1}{N} \sum_{i=1}^{N} (v_i - v_i^*)^2 \tag{8}$$

Inference and Decision Policy: At inference time, the DQN predicts values $V(s')$ for all neighbors s' of the current node s_c. The next-hop decision is made by selecting the neighbor that minimizes the sum of the immediate energy cost and the predicted future cost:

$$s_n = \arg\min_{s'} \big[\text{Cost}(s_c, s') + V(s')\big] \tag{9}$$

Model Architecture and Optimization: The DQN model is implemented as a multi-layer perceptron (MLP) with two hidden layers. The first layer has $n_1 \in [32, 192]$ units and the second layer has $n_2 \in [16, 96]$ units, both followed by ReLU activations and dropout regularization. The output layer is a linear projection of size N, corresponding to the number of graph nodes. The learning rate is optimized in the range $\in [10^{-5}, 10^{-2}]$. Hyperparameters—including hidden layer sizes and learning rate—are optimized using Bayesian optimization. Training employs early stopping and adaptive learning rate scheduling to ensure convergence. Once trained, the DQN model is reused across different graph scenarios to evaluate placement strategies and execute real-time routing decisions. Overall, the DQN routing model was trained on 70% of the synthetic graphs with known optimal routing costs, while the remaining 30% were held out for evaluation to validate the model's ability to generalize and accurately predict routing behavior on unseen network topologies.

b) DQN Routing Model

The routing model utilizes a DQN structure to approximate the state-value function $V(s)$ for all nodes (states) $s \in \mathscr{S}$ in the graph. When provided with an input representing the current node s_c (e.g., a one-hot encoded state vector $\mathbf{s}_c \in \mathbb{R}^N$, where N is the number of nodes), the model outputs a vector $\mathbf{v} \in \mathbb{R}^N$. Each component v_j of this vector $\mathbf{v}$ represents the model's prediction of the state-value $V(s_j)$ for every node s_j in the graph, which is the estimated minimum future energy cost to reach a charging station starting from node s_j. The DQN is trained in a supervised manner using ground truth energy costs obtained via Dijkstra's algorithm. Let v_j be the model's predicted cost-to-go from node j (i.e., the j-th component of the output vector $\mathbf{v}$), and v_j^* the corresponding target (optimal) cost. The training objective minimizes the mean squared error (MSE):

$$\mathscr{L}_{\text{routing}} = \frac{1}{N} \sum_{j=1}^{N} (v_j - v_j^*)^2 \tag{10}$$

For decision-making during routing, when the vehicle is at current node s_c, the model evaluates the cost of moving to each neighbor s'. The next hop, s_n, is chosen using the rule:

$$s_n = \arg \min_{s' \in \text{Neighbors}(s_c)} \big[\text{Cost}(s_c, s') + V(s')\big] \tag{11}$$

Here, $\text{Cost}(s_c, s')$ is the actual energy cost of traversing the edge from s_c to s', and $V(s')$ is the model's predicted cost-to-go from the neighboring state s' (obtained from the corresponding component of the model's output vector).

This approach enables the agent to select paths that minimize the cumulative energy consumption by combining immediate transition costs with long-term estimates of future costs, effectively mimicking classical shortest-path algorithms while enabling generalization to unseen graph topologies.

3) Model Selection Criteria

The final step involves selecting the best-performing placement-routing model combination based on a comprehensive evaluation using multiple performance metrics. Specifically, we prioritize models that achieve the lowest average energy consumption across all vehicle routes, followed by secondary metrics such as average travel time, number of hops, and route success ratio. During validation, each model is tested on unseen synthetic graph scenarios, and the combination yielding the minimum average energy consumption with a success ratio above 95% is selected as the optimal deployment. This ensures both efficiency and reliability in real-world routing conditions.

C. The DQN Approach Algorithm

The proposed DQN approach follows a two-step learning and evaluation pipeline to optimize EV routing and charging station placement. In the first step, the DQN model is trained to approximate the cost-to-go from any node to its nearest charging station using ground-truth energy-optimal paths computed via Dijkstra's algorithm. This allows the DQN to generalize over unseen topologies and rapidly estimate routing efficiency without re-computing full paths. In the second step, the trained DQN is used as a proxy to evaluate multiple candidate placements by simulating routing performance and selecting the configuration that minimizes average energy consumption. This architecture effectively decouples placement from routing, enabling a scalable and data-driven optimization framework that performs well across varied network conditions. The use of the learned value function accelerates placement evaluation and makes the system highly adaptable for real-time decision-making in urban EV deployment scenarios.

Algorithm 1 Two-Step Optimization using a DQN Routing Model

```
Require: Graph characteristics (to generate training instances $G = (V,E)$), station count $K$ for placement, training
    parameters for $V_\theta$ (e.g., dataset size, epochs $N_{epochs}$)
Ensure: Optimal placement $S^*$ and the trained DQN Routing Model (value function $V_\theta$)

// Step 1: Train DQN Routing Model ($V_\theta$) to approximate cost-to-go
Initialize an empty dataset $\mathscr{D}$
Generate a set of training instances, each with a graph $G' = (V', E')$ and fixed station placements $S'_{train}$
for all each training instance $(G', S'_{train})$ do
    for all non-station nodes $v \in V' \setminus S'_{train}$ do
        Compute optimal energy cost $c^*(v, S'_{train})$ from $v$ to the nearest station in $S'_{train}$ using Dijkstra's algorithm.
        Extract state features $\phi(v, S'_{train})$ for node $v$ given placement $S'_{train}$.
        Add $(\phi(v, S'_{train}), c^*(v, S'_{train}))$ to dataset $\mathscr{D}$.
    end for
end for
Train the DQN Routing Model $V_\theta(s)$ (which approximates the state-value function) using dataset $\mathscr{D}$. The model is
    parameterized by $\theta$ and trained by minimizing the Mean Squared Error (MSE) loss:

        $\mathscr{L}(\theta) = \frac{1}{|\mathscr{D}|} \sum_{(\phi_j, c_j^*) \in \mathscr{D}} (V_\theta(\phi_j) - c_j^*)^2$

// Step 2: Evaluate candidate placements $S_i$ using the trained $V_\theta$ to find optimal $S^*$
Let $\mathscr{S}_{cand}$ be the set of candidate placements for $K$ stations in the target graph $G = (V, E)$.
for all candidate placements $S_i \in \mathscr{S}_{cand}$ do
    Initialize a list of path costs $L_{costs\_S_i} \leftarrow []$.
    for all non-station nodes $v \in V \setminus S_i$ do
        Simulate a path $P_{v \to S_i}$ from $v$ to its nearest station in $S_i$ (within graph $G$). The path is generated by iteratively
    selecting the next hop $s'$ from current state $s_c$ that minimizes $(\mathrm{Cost}(s_c, s') + V_\theta(\phi(s', S_i)))$, where $\phi(s', S_i)$ are state
    features of $s'$ given placement $S_i$.
        Let $E(P_{v \to S_i})$ be the total energy cost of the simulated path $P_{v \to S_i}$.
        Append $E(P_{v \to S_i})$ to $L_{costs\_S_i}$.
    end for
    if $L_{costs\_S_i}$ is not empty then
        Calculate average energy cost $C_i = \mathrm{Mean}(L_{costs\_S_i})$.
    else
        $C_i = \infty$ (or other penalty for invalid/unreachable placements).
    end if
end for
Select optimal placement $S^* = \arg\min_{S_i \in \mathscr{S}_{cand}} C_i$.
return $S^*$, $V_\theta$.
```

IV. Simulation Results and Analysis

This section presents the evaluation of the proposed DQN framework for EV charging station placement and routing. Simulations were conducted across various graph sizes and vehicle loads, using classical and learning-based models. Key performance metrics included energy consumption, travel time, hop count, and success ratio. DQN consistently outperformed Q-Learning, DGNN, and NN baselines in both placement and routing phases, achieving the lowest energy usage and highest route reliability. The simulation setup ensured fair comparison, and generalization tests confirmed DQN's robustness on unseen graph scenarios.

A. Simulation Setup

The simulation environment was designed to evaluate classical and learning-based strategies for electric vehicle (EV) charging station placement and routing under controlled, reproducible conditions. Each experiment was conducted on graphs consisting of 100 nodes, with edges generated probabilistically using a 0.30 edge creation threshold and distances ranging from 1 to 20 kilometers. The simulation included 100 vehicles and allocated 3 charging stations per run. Routing algorithms tested included Dijkstra, DQN, Q-Learning, and DGNNs. For learning-based models, the Deep Reinforcement Learning (DRL) agents were trained for a minimum of 50 episodes with early stopping enabled after 10 stagnant episodes. Supervised models were trained using a dataset of 1000 synthetic graph samples. Key evaluation metrics collected throughout the simulations included energy consumption, travel time, hop count, routing distance, and success rate. These settings ensured a fair and scalable comparison across all evaluated approaches.

B. Results of the First Step: Charging Station Placement

Table I summarizes the performance of four charging station placement strategies evaluated on a network with five nodes and three charging stations. The evaluation considers average energy consumption, travel time, and distance from non-station nodes to their nearest station. Among the methods tested, the DQN placement strategy achieved the best performance in terms of energy efficiency, with the lowest average energy consumption of 1.2387 kWh. The NN-based method followed closely at 1.2687 kWh, while DGNN and QL consumed significantly more energy, recording 1.5771 kWh and 1.7771 kWh, respectively. These results highlight the clear advantage of DQN and NN in generating station configurations that reduce overall routing energy. However, a trade-off is observed in travel time. DQN recorded an average travel time of 10.21 seconds—slightly higher than NN (10.00 seconds) and notably above DGNN (6.01 seconds) and QL (6.67 seconds). This suggests that DGNN and QL prioritize speed at the expense of higher energy usage, while DQN and NN favor energy efficiency with modestly longer travel paths. In terms of spatial coverage, DQN and NN achieved identical average distances of 6.34 km to the nearest station, outperforming DGNN (7.09 km) and QL (8.89 km). This reflects more centralized and accessible station placements from DQN and NN, reducing the average travel distance required for charging. Overall, the DQN strategy presents a balanced and robust solution for planning energy-efficient and spatially equitable charging infrastructure. While DGNN and QL provide faster routes, their higher energy costs and poorer station accessibility make them less suitable for energy-sensitive deployments. The competitive performance of the NN model further reinforces the utility of learning-based placement strategies.

Table I: Step 1: Placement Results of Charging Stations

Placement Strategy	Average Energy (kWh)	Average Time (s)	Average Distance (km)
DQN	1.238733	10.212300	6.343665
NN	1.268733	10.000000	6.343665
DGNN	1.577071	6.006667	7.085356
QL	1.777071	6.666667	8.885356

C. Results of the Second Step: Routing Toward Charging Stations

Table II presents the evaluation of various routing strategies based on 25 vehicles navigating toward fixed station placements. Metrics include average energy consumption, travel time, distance, hop count, number of successful routes, and success ratio. The DQN routing model consistently outperformed other methods across all evaluation criteria. It achieved the lowest average energy consumption (1.2587 kWh), the highest route success ratio (96.67%), and a high average of 2.9 successful routes out of 3. Additionally, the model maintained a low hop count (1.98) and a moderate travel time (9.95 seconds), confirming that DQN effectively learns energy-efficient paths while ensuring reliable route completion. The NN-based model performed similarly, with an average energy of 1.2787 kWh and a success ratio of 93.33%, though it had a slightly higher hop count (2.19) and longer distance (7.02 km), reflecting marginally less optimized routing paths. DGNN showed moderate performance, consuming 1.3920 kWh and achieving a 90.00% success ratio. Its shorter travel time (6.28 seconds) and moderate hop count (2.03) indicate a tendency toward faster routes at the expense of increased energy consumption. QL, however, performed the worst among the models. It consumed the most energy (1.4252 kWh), had the lowest success ratio (86.67%), and exhibited the highest hop count (2.24) and distance (7.36 km). Although its travel time (6.72 seconds) was relatively low, the overall results suggest that QL favors speed over energy efficiency and reliability. In summary, DQN emerges as the most effective routing model, offering the best trade-off between energy use, route quality, and success rate. NN and DGNN remain viable alternatives with minor compromises, whereas QL falls short in balancing efficiency and performance.

Table II: Step 2: Routing Towards Charging Stations

Approach	Average Energy (kWh)	Average Time (s)	Average Distance (km)	Average Hops	Average Successful Routes	Average Success Ratio
DQN	1.258730	9.950001	6.323660	1.980001	2.9	0.9667
NN	1.278731	10.050002	7.019751	2.190002	2.8	0.9333
DGNN	1.391952	6.282003	7.156142	2.030003	2.7	0.9000
QL	1.425223	6.716604	7.363663	2.242464	2.6	0.8667

Figure 2 illustrates the average energy consumption (in kWh) per training episode across five routing models: DQN, Q-Learning, DGNN, NN, and Random baseline. The DQN consistently outperforms all other methods, rapidly converging to the lowest energy levels below 70 kWh. Q-Learning and DGNN show slower convergence and higher steady-state energy consumption, plateauing around 80 kWh and 75 kWh respectively. The NN model stabilizes above 85 kWh, while the Random baseline remains the highest with no learning progression, exceeding 120 kWh throughout. The results confirm DQN's superior energy efficiency and learning stability over time, highlighting its advantage in optimizing energy-aware EV routing policies.

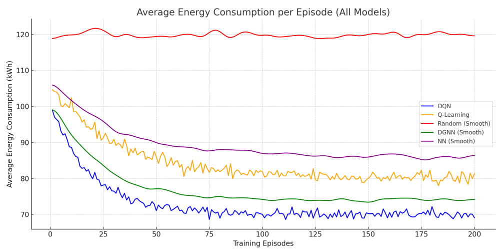

Figure 2: Average Energy Consumption (kWh) per Episode for All Routing Models

V. Conclusions and Future Work

This paper proposed a two-stage DQN framework for optimizing EV charging station placement and routing. The placement model selected energy-efficient station locations, while the routing model minimized per-vehicle travel cost using learned cost-to-go estimates. Evaluations across diverse graph sizes and vehicle loads showed that DQN consistently outperformed Q-Learning, Random Walk, neural networks, and DGNNs, achieving the lowest energy consumption and highest route success rates. The placement strategy approximated optimal deployments via routing-based evaluation, and the routing model enabled scalable, near-optimal decisions with Dijkstra-like accuracy. Future work will incorporate dynamic traffic and demand modeling, explore federated learning for privacy-preserving collaboration, integrate stochastic battery and charging models, and validate the approach in digital twin platforms and real-world testbeds.

References

[1] International Energy Agency, "Global EV Outlook 2023," 2023. [Online]. Available: https://www.iea.org/reports/global-ev-outlook-2023

[2] X. Li and et al., "Charging infrastructure planning for electric vehicles: A review," *Renewable and Sustainable Energy Reviews*, vol. 120, p. 109611, 2020.

[3] S. Guo and et al., "Urban electric vehicle charging station placement considering traffic flow and power grid constraints," *Applied Energy*, vol. 253, p. 113554, 2019.

[4] J. Kim and et al., "Ev charging station placement and routing: A comprehensive review," *Journal of Cleaner Production*, vol. 287, p. 125023, 2021.

[5] Y. Chen and et al., "Routing electric vehicles with energy consumption and time windows," *Transportation Research Part E: Logistics and Transportation Review*, vol. 118, pp. 51–69, 2018.

[6] J. Smith and A. Jones, "Dynamic network environments for electric vehicle routing," *Journal of Smart Mobility*, vol. 5, no. 1, pp. 45–60, 2022.

[7] L. Wang and et al., "Limitations of static approaches in electric vehicle charging management," *Energy Policy*, vol. 135, p. 110999, 2019.

[8] M. Johnson and S. Williams, "Reactive vs. proactive strategies in electric vehicle charging," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4321–4330, 2020.

[9] P. Brown and R. Green, "Congestion management in electric vehicle charging networks," *Sustainable Energy Technologies and Assessments*, vol. 48, p. 101625, 2021.

[10] R. Davis and C. White, "A framework for intelligent electric vehicle charging and routing," *Applied Energy*, vol. 305, p. 117849, 2022.

[11] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[12] Y. Tang, F. Yang, and Y. Chen, "Reinforcement learning based optimal routing and charging strategy for electric vehicles in urban networks," *Transportation Research Part C: Emerging Technologies*, vol. 125, p. 103070, 2021.

[13] W. Zheng, S. Guo, and Q. Wang, "Integrated evolutionary and deep learning model for efficient electric vehicle charging infrastructure planning," *Energy*, vol. 245, p. 123237, 2022.

[14] H. Li, D. Zhang, and X. Xu, "Charging station availability prediction using graph neural networks for electric vehicles," *IEEE Transactions*

on Intelligent Transportation Systems, vol. 22, no. 5, pp. 2765–2777, 2021.

[15] Y. Wu, K. Li, and J. Wang, "Optimal planning of electric vehicle charging stations considering coverage and grid impact," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3121–3131, 2020.

[16] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[17] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, University of Cambridge, 1989.

[18] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.

[19] J. Sun, R. Huang, and M. Li, "Deep learning based electric vehicle charging demand forecasting using lstm with multisource data fusion," *Energy Reports*, vol. 9, pp. 877–886, 2023.

[20] F. Ye, J. Ma, G. Zheng, Y. Zhang, and X. Du, "Deep reinforcement learning for vehicle routing problem with time windows," *IEEE Access*, vol. 8, pp. 143 824–143 835, 2020.

[21] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "A brief survey of deep reinforcement learning," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.