Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Benchmarking Adaptive EV Charging Recommendation Under Feasibility Constraints and Non-Stationary Infrastructure

vibhor joshi

vibhor.joshi@iiitg.ac.in

indian Institute of information technology guwahati https://orcid.org/0009-0003-1074-0774

Research Article

Keywords: Electric vehicle charging, recommendation systems, benchmarking, contextual bandits, reinforcement learning, feasibility constraints, non-stationary environments, dynamic pricing, charging station availability, adaptive decision-making, explainable recommendations

Posted Date: April 15th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9415800/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Benchmarking Adaptive EV Charging Recommendation Under Feasibility Constraints and Non-Stationary Infrastructure

Abstract

The rapid adoption of electric vehicles (EVs) has intensified the need for charging station recommendation systems that can adapt continuously to changing infrastructure conditions. In practice, charging environments are highly dynamic: station availability fluctuates; prices vary over time, and charging capabilities differ widely across locations. However, many existing approaches assume static infrastructure attributes or optimize a single objective in isolation, limiting their effectiveness in real-world deployments where robust adaptation under uncertainty is essential.

In this work, we study EV charging recommendations as a constraint-aware online decision problem and present a unified benchmarking framework for evaluating learning-based recommendation policies under both static and dynamic regimes. Using a large-scale global charging infrastructure dataset (1.5M stations) and a real-world time-series availability and pricing dataset, we compare contextual bandit and reinforcement learning approaches across multiple dimensions, including relevance, feasibility, regret, adaptability, and recovery from exogenous shocks.

Our results show that learning-based methods significantly reduce constraint violations and cumulative regret compared to heuristic baselines in static settings, while neural contextual bandits achieve the strongest feasibility–utility trade-off. Under dynamic conditions, we observe clear differences in adaptation speed and shock recovery across algorithms, indicating that effective EV charging recommendation requires not only feasibility-aware reward design, but also robustness to non-stationary pricing and availability and the preservation of a meaningful learning signal during disruptive events.

Keywords: Electric vehicle charging; recommendation systems; benchmarking; contextual bandits; reinforcement learning; feasibility constraints; non-stationary environments; dynamic pricing; charging station availability; adaptive decision-making; explainable recommendations

1 Problem Statement

Although the number of public charging stations has increased substantially, EV users still face significant uncertainty when selecting where to charge. Charging networks are fragmented across providers, pricing varies widely even within the same city, and real-time availability is often unreliable or incomplete. A station that appears suitable on a map may be offline, congested, or incompatible with the user's charging requirements. As a result, users must make decisions based on partial and frequently outdated information.

Most existing recommendation systems rely on fixed ranking rules or static optimisation objectives, such as minimizing distance or price. These approaches do not adapt to changing conditions, do not learn user preference trade-offs, and provide no guarantees that recommended stations will remain feasible at the time of arrival. Reinforcement learning–based methods have been explored,

but many assume stable rewards or focus on a single performance metric, without explicitly accounting for feasibility violations or responsiveness to infrastructure disruptions.

Rather than proposing a new optimisation algorithm, this work introduces a benchmarking framework for evaluating learning-based EV charging recommendation policies. The framework separates feasibility enforcement from preference learning, supports both static and dynamic infrastructure regimes, and enables controlled comparison of multiple algorithms under identical constraints, feedback budgets, and environmental conditions.

2 Related Work

Early research on EV charging station recommendation primarily framed the problem as a routing or optimisation task, where the objective was to minimise travel distance or charging cost given fixed infrastructure attributes. More recent work has explored learning-based formulations, particularly reinforcement learning, to capture user preferences and spatiotemporal charging demand. Representative studies, such as those by Zhang et al., apply reinforcement learning to charging station selection and demonstrate improvements over static heuristics in simulated environments.

A parallel line of research focuses on dynamic pricing and load balancing in EV charging networks. Studies such as those by Li et al. investigate non-stationary pricing schemes, often using reinforcement learning or control-theoretic approaches to align charging demand with grid objectives. These works highlight the importance of adapting to temporal fluctuations in demand and cost, particularly in urban charging scenarios.

Contextual bandits and reinforcement learning provide the methodological foundation for adaptive recommendation systems. Linear contextual bandits, such as LinUCB, have been widely adopted due to their sample efficiency and interpretability, making them suitable baselines for static recommendation settings. Extensions based on Thompson Sampling further improve exploration through Bayesian posterior sampling. More recently, neural contextual bandits, including NeuralUCB and bootstrap-based approaches, have been proposed to capture non-linear reward structures while retaining bandit-style learning dynamics.

3 Methodology

3.1 Problem Definition

We formulate EV charging station recommendation as a constraint-aware online decision problem over a finite interaction horizon T. At each step $t \in \{1, \ldots, T\}$, a user query q_t is observed, defining the context

$$c_t = \{\ell_u, \mathcal{C}_t, \mathcal{S}_t\}, \tag{1}$$

where the user location is given by $\ell_u = (\text{lat}_u, \text{lon}_u)$, the hard operational constraints are denoted by $\mathcal{C}_t = (d_{\max}, p_{\min}, \kappa)$, and the observable infrastructure state is represented by $\mathcal{S}_t = \{\rho_t(a), \sigma_t(a)\}_{a \in \mathcal{A}}$. Here, $\rho_t(a)$ denotes the time-dependent charging price and $\sigma_t(a) \in \{0, 1\}$ encodes station availability. Static station metadata is stored as an immutable set

$$\mathcal{D}_{\text{static}} = \{(\ell_a, P_a, \kappa_a)\}_{a \in \mathcal{A}}, \tag{2}$$

where ℓ_a denotes the geographic coordinates of station a, P_a is its maximum deliverable charging power (kW), and κ_a specifies its connector type. Dynamic infrastructure signals are ingested as time-indexed snapshots $\{\mathcal{S}_t\}_{t=1}^{T}$, ensuring that static and dynamic decision regimes remain separable.

At each interaction step, a feasible candidate action set $\mathcal{A}_t \subseteq \mathcal{A}$ is constructed via geographic filtering and hard constraint enforcement. A station $a \in \mathcal{A}$ is admitted only if

$$d(\ell_u, a) \leq d_{\max}, \quad P_a \geq p_{\min}, \quad \sigma_t(a) = 1, \tag{3}$$

where $d(\ell_u, a)$ is computed using the Haversine distance. Thus, the feasible candidate set is defined as

$$\mathcal{A}_t = \{a \in \mathcal{A} : d(\ell_u, a) \leq d_{\max},\, P_a \geq p_{\min},\, \sigma_t(a) = 1\}\,. \tag{4}$$

This feasibility-first design ensures that infeasible charging recommendations are excluded rather than treated as suboptimal actions. For each candidate $a \in \mathcal{A}_t$, a feature representation is constructed as

$$x_t(a) = \left[\hat{d}_t(a), \hat{\rho}_t(a), \hat{P}_a, \varphi(t)\right], \tag{5}$$

where $\hat{d}_t(a)$, $\hat{\rho}_t(a)$, and $\hat{P}_a$ denote normalized distance, price, and power attributes, and $\varphi(t)$ optionally encodes temporal context. The resulting policy input is

$$X_t = \{x_t(a)\}_{a \in \mathcal{A}_t}. \tag{6}$$

A pluggable decision policy π_θ is then applied under a shared interface, selecting an action according to

$$a_t = \pi_\theta(c_t, X_t). \tag{7}$$

After selection, a scalar feedback signal is observed as

$$r_t = U(c_t, a_t), \tag{8}$$

implemented as a transparent utility proxy derived from observable station attributes. A typical reward form is given by

$$r_t = w_1 \hat{P}(a_t) + w_2(1 - \hat{\rho}_t(a_t)) + w_3(1 - \hat{d}_t(a_t)) - \lambda \cdot \mathbb{1}[a_t \notin \mathcal{A}_t]. \tag{9}$$

Policy parameters are updated online via

$$\theta_{t+1} = \pi.\text{update}(\theta_t, c_t, a_t, r_t), \tag{10}$$

and the interaction history is logged as

$$\mathcal{L} = \{(c_t, a_t, r_t)\}_{t=1}^{T}. \tag{11}$$

Performance is evaluated using relevance $(1/T)\sum_{t=1}^{T} r_t$, constraint violation rate $(1/T)\sum_{t=1}^{T} \mathbb{1}[a_t \notin \mathcal{A}_t]$, and oracle regret, where

$$r_t^* = \max_{a \in \mathcal{A}_t} U(c_t, a), \tag{12}$$

$$\Delta_t = r_t^* - r_t, \tag{13}$$

$$R_T = \sum_{t=1}^{T} \Delta_t. \tag{14}$$

By repeating the interaction loop under controlled query budgets and, in the dynamic regime, under exogenous shocks to $\rho_t(a)$ and $\sigma_t(a)$, systematic benchmarking of feasibility, adaptability, robustness, and computational cost is enabled. Since all policies share identical inputs (c_t, X_t), constraints $\mathcal{C}_t$, and feedback r_t, observed performance differences are directly attributable to learning behaviour rather than implementation artefacts.

3.2 System Architecture and Modular Benchmarking Framework

The constraint-aware benchmarking framework was structured into four layers (Fig. 1), ensuring that all evaluated algorithms received identical inputs, constraints, and feedback signals. The modular design enabled fair comparison across learning paradigms while isolating the effects of learning dynamics from implementation artefacts.

3.2.1 Layer 1: Data Engineering and State Management

Immutable station metadata was strictly separated from time-indexed infrastructure snapshots. Static station attributes $\mathcal{D}_{\text{static}} = \{(\ell_a, P_a, \kappa_a)\}_{a\in\mathcal{A}}$ were loaded once at initialization and remained constant throughout the experiment. Dynamic infrastructure signals $\mathcal{S}_t = \{\rho_t(a), \sigma_t(a)\}_{a\in\mathcal{A}}$ were ingested as discrete time-indexed snapshots $\{\mathcal{S}_t\}_{t=1}^{T}$. This temporal separation prevented inadvertent leakage of future availability or pricing information into the policy state, ensuring that algorithms operated only on information available at decision time t.

3.2.2 Layer 2: Feasible Candidate Generation and Feature Construction

At each interaction step t, a feasible candidate action set $\mathcal{A}_t \subseteq \mathcal{A}$ was constructed via sequential constraint enforcement:

$$a \in \mathcal{A}_t \Leftrightarrow d(\ell_u, a) \leq d_{\max}, \quad P_a \geq p_{\min}, \quad \sigma_t(a) = 1. \tag{15}$$

This feasibility-first filtering ensured that infeasible actions were excluded from the candidate pool before policy selection, rather than penalized post-hoc. For all admitted stations $a \in \mathcal{A}_t$, normalized feature vectors were constructed:

$$x_t(a) = \left[\hat{d}_t(a), \hat{\rho}_t(a), \hat{P}_a, \varphi(t)\right], \tag{16}$$

where $\hat{d}_t(a)$, $\hat{\rho}_t(a)$, and $\hat{P}_a$ denote normalized distance, price, and power attributes respectively, and $\varphi(t)$ encodes optional temporal context. Normalization was applied to ensure feature scaling remained comparable across heterogeneous stations and time periods.

3.2.3 Layer 3: Shared Policy Interface

All learning algorithms were instantiated behind a common decision interface. The policy selected actions according to:

$$a_t = \pi_\theta\left(c_t, \{x_t(a)\}_{a\in\mathcal{A}_t}\right), \tag{17}$$

where π_θ denotes the parametric policy and θ represents trainable parameters. This unified interface ensured that algorithmic variation arose exclusively from learning dynamics and exploration strategies, rather than from differential access to contextual information or candidate sets.

3.2.4 Layer 4: Simulation Loop and Exogenous Shock Injection

After action selection, a scalar utility-based reward was computed as:

$$r_t = U(c_t, a_t), \tag{18}$$

Policy parameters were updated online via:

$$\theta_{t+1} = \pi.\text{update}(\theta_t, c_t, a_t, r_t), \tag{19}$$

Regret and feasibility violation metrics were computed and logged at each step. Exogenous shocks were injected after a predefined timestamp t_s to evaluate robustness under abrupt infrastructure disruption. The simulation loop was repeated under controlled query budgets and varying shock magnitudes.

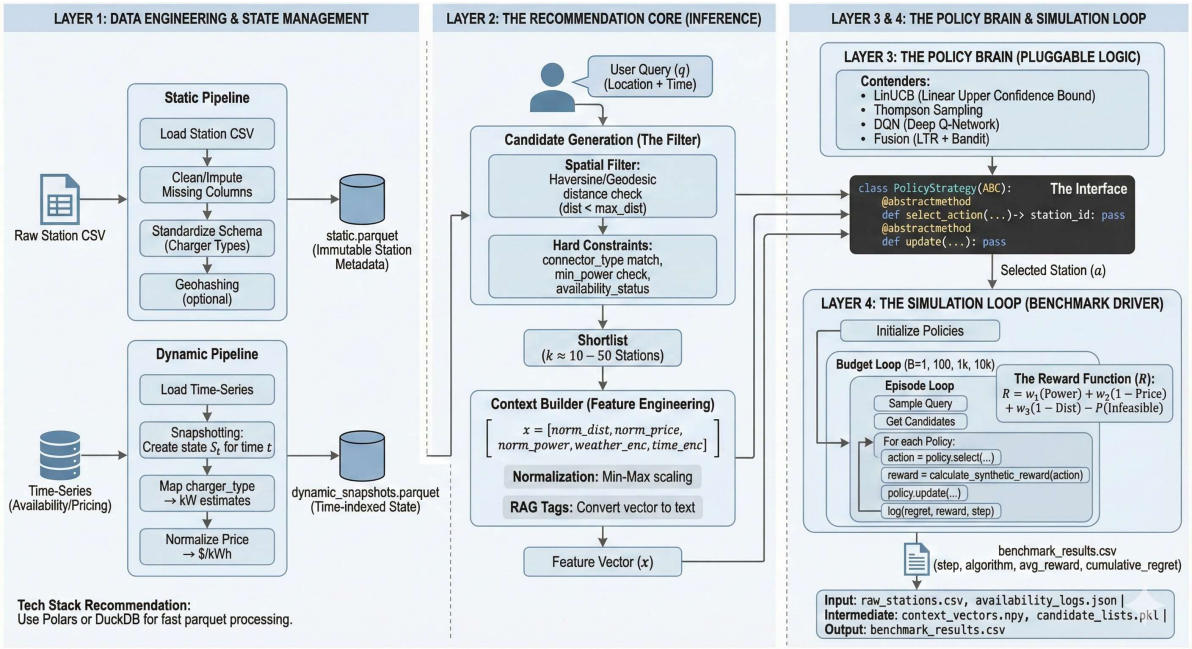


Figure 1: System architecture for constraint-aware EV charging recommendation. The modular four-layer design ensures fair algorithmic comparison by separating data management (Layer 1), feasibility enforcement (Layer 2), policy interfaces (Layer 3), and simulation mechanics (Layer 4).

Figure 1 illustrates the complete system architecture. The architecture is organized as a modular pipeline where each layer has a distinct responsibility, ensuring that algorithms are evaluated under identical conditions without implementation bias. Layer 1 manages the separation of immutable station metadata from dynamic infrastructure snapshots, preventing information leakage across temporal boundaries. Layer 2 enforces hard feasibility constraints through geographic filtering and power requirements, constructing the set of admissible stations before any learning policy is consulted. Layer 3 provides a unified decision interface where all algorithms, regardless of internal structure, receive the same feature vectors and must produce an action selection. Layer 4 closes the feedback loop by computing rewards, updating parameters, and logging performance metrics. This four-layer decomposition is fundamental to the fairness of the benchmarking, as it ensures that performance differences are attributable solely to learning behaviour rather than to variations in feature access, constraint handling, or information timing.

3.3 Benchmarking Protocol and Algorithm Conceptualization

3.3.1 Interaction Log and Benchmarking Mathematics

A comprehensive interaction log was constructed at each experiment:

$$\mathcal{L} = \{(t, a_t, r_t, \text{regret}_t, \nvDash_{\text{violation}})\}_{t=1}^{T}, \tag{20}$$

where each tuple recorded the decision step, selected action, observed reward, incurred regret, and whether the selected action violated feasibility constraints. This log enabled systematic benchmarking across query budgets, shock magnitudes, and infrastructure regimes.

Two design principles underlay this logging framework. First, temporal snapshotting of dynamic signals was required to model evolving infrastructure states without introducing information leakage. Second, feature normalization was required to ensure that learned policies operated on comparable scales across heterogeneous stations, time periods, and attribute types.

3.3.2 Algorithm Paradigm Comparison

The selected algorithms represented distinct learning paradigms suited to different infrastructure regimes. LinUCB and Thompson Sampling were formulated as linear contextual bandits learning stationary reward models from immediate feedback, and were therefore best suited to static regimes where station attributes and user preferences remained constant. Neural contextual bandits extended this setting by modelling non-linear reward structure through bootstrapped neural networks, enabling partial adaptability under mildly dynamic conditions while maintaining bandit-style exploration.

Deep reinforcement learning methods such as DQN and PPO were evaluated under dynamic non-stationarity, where sequential dependence and recovery from evolving pricing or availability shifts were required. Greedy and random baselines were treated as non-adaptive heuristics, ignoring temporal congestion dynamics and feedback-driven learning signals.

Constrained single-agent variants such as CDQN and CPPO incorporated feasibility-aware reinforcement learning updates but remained limited in multi-agent interaction settings due to single-agent formulation. Multi-agent reinforcement learning approaches such as MADDPG-ATT, MASTER, and RLCHARGE provided increased flexibility through explicit agent coupling and communication, although deployment complexity was increased due to centralized training overhead and computational requirements.

The modular constraint-aware benchmarking framework enabled these paradigms to be evaluated under identical feasible candidate sets and shared feedback signals, supporting controlled comparison of feasibility compliance, adaptability, and robustness under both static and dynamic infrastructure regimes.

3.4 Evaluation Metrics and Performance Assessment

Performance was not assessed using aggregate reward alone, but through complementary metrics capturing operational validity, user-facing utility, learning efficiency, and robustness to disruption.

3.4.1 Relevance and Preference Learning

Relevance was measured as the expected instantaneous reward:

$$\text{Rel} = \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} r_t\right], \tag{21}$$

capturing the average utility achieved by the recommendation policy across the interaction horizon. Higher relevance indicated stronger alignment between recommended stations and the underlying utility function defined by the experimental design.

3.4.2 Feasibility Compliance

Constraint violation rate was computed as:

$$\text{ViolRate} = \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}[a_t \notin \mathcal{A}_t], \tag{22}$$

quantifying the fraction of interactions in which the policy selected an infeasible action. This metric directly assessed the effectiveness of the constraint-aware framework.

3.4.3 Regret and Optimality Gap

Oracle regret was computed relative to the optimal feasible action at each step:

$$r_t^* = \max_{a \in \mathcal{A}_t} U(c_t, a), \tag{23}$$

$$\Delta_t = r_t^* - r_t, \tag{24}$$

$$R_T = \sum_{t=1}^{T} \Delta_t, \tag{25}$$

capturing the cumulative opportunity cost of the learning algorithm relative to perfect knowledge of feasible rewards.

3.4.4 Robustness Under Exogenous Shocks

To evaluate adaptability under non-stationary infrastructure, controlled shocks were injected after a predefined step t_s. Pricing was perturbed multiplicatively:

$$\rho_t(a) \leftarrow \alpha \rho_t(a), \quad t \geq t_s, \tag{26}$$

and a fraction of stations was forced offline via $\sigma_t(a) \leftarrow 0$. Policy resilience was measured through recovery time:

$$\tau_{\text{rec}} = \min\left\{t - t_s : \bar{r}_t \geq 0.95 \bar{r}_{t_s}\right\}, \tag{27}$$

where $\bar{r}_t$ denotes a rolling window average of rewards. Recovery time captured the number of interactions required for the policy to return to pre-shock performance levels.

3.4.5 Benchmarking Regime Specification

Contextual bandit algorithms were evaluated primarily under the static regime with immediate-reward feedback, while reinforcement learning methods were stress-tested under the dynamic regime with sequential shocks and non-stationary pricing. Since all policies operated on identical feasible candidate sets and shared feature inputs, observed performance differences directly reflected algorithmic adaptability, feasibility compliance, and robustness rather than artifacts of implementation, hyperparameter selection, or inconsistent information access.

3.5 Dataset Section

The proposed charging recommendation formulation was evaluated under both static and dynamic infrastructure regimes. In the static regime, station attributes were assumed fixed across interactions, resulting in independent decision steps consistent with contextual bandit evaluation. In contrast, the dynamic regime incorporated time-varying pricing and availability signals, inducing non-stationarity and requiring policies capable of adaptation and recovery under evolving system conditions.

A real-world charging station metadata table was used as the core dataset, where each record corresponds to a station $a \in A$. The dataset contained the operational attributes required for feasibility filtering and utility computation, as summarised in Table 1. Formally, the static station collection was represented as

$$\mathcal{D}_{\text{static}} = \{(\ell_a, P_a, \rho_a, \sigma_a)\}_{a \in A}, \tag{28}$$

where $\ell_a = (\text{lat}_a, \text{lon}_a)$ denotes geographic coordinates, P_a is the maximum charging power, ρ_a is the energy cost per kWh, and $\sigma_a \in \{0, 1\}$ encodes operational status. These attributes were used to construct the feasible candidate action set $\mathcal{A}_t$ under hard constraints, including maximum travel distance $d_{\max}$ and minimum charging power $p_{\min}$. Candidate feasibility was enforced by admitting only stations satisfying

$$a \in \mathcal{A}_t \Leftrightarrow d(\ell_u, a) \leq d_{\max}, \quad P_a \geq p_{\min}, \quad \sigma_t(a) = 1. \tag{29}$$

Since real-world charging datasets often exhibit heterogeneous naming conventions across sources, a schema harmonisation procedure was applied prior to benchmarking. Column headers were mapped into a canonical representation, and missing operational values were filled using conservative defaults to ensure robustness of preprocessing. This standardisation prevented inconsistencies during candidate generation and ensured reproducibility across datasets.

For evaluation, realistic user queries were generated by sampling anchor stations and perturbing their coordinates with small Gaussian noise. At each interaction step, the candidate action set was formed by selecting the k nearest feasible stations ($k \approx 30$–50) according to the Haversine distance metric,

$$d(\ell_u, a) = 2R \arcsin \sqrt{h(\ell_u, \ell_a)}, \tag{30}$$

thereby avoiding degenerate infeasibility and ensuring meaningful reward variation across actions. Feature vectors $x_t(a)$ were then constructed from normalised distance, power, and pricing attributes, forming the sole input to all evaluated policies.

In the dynamic regime, an additional time-indexed infrastructure stream was incorporated as

$$\mathcal{S}_t = \{\rho_t(a), \sigma_t(a)\}_{a \in A}, \tag{31}$$

capturing evolving electricity tariffs and station availability over time. Exogenous shocks were simulated after a predefined step t_s through multiplicative price surges,

$$\rho_t(a) \leftarrow \alpha \rho_t(a), \quad t \geq t_s, \tag{32}$$

and forced station outages, enabling systematic evaluation of robustness and recovery behaviour under abrupt infrastructure disruption. This dataset construction ensured that all algorithms were evaluated under identical station inputs, feasibility constraints, and evolving infrastructure signals, thereby supporting fair comparison across static and dynamic settings.

Table 1: Dataset usage and attributes

Attribute	Static dataset	Dynamic dataset
Latitude / Longitude	✓	✓
Max power (kW)	✓	mapped
Cost per kWh	✓	✓(time-varying)
Availability status	✓	✓(time-varying)
Temporal signals	×	✓

3.6 Algorithms

All algorithms in this study share a common interaction loop. At each step, the system constructs a candidate set based on geographic proximity and hard constraints, encodes each candidate into a feature representation, and applies a policy to select a charging station. After observing feedback in the form of a synthetic reward, the policy is updated, and relevant metrics are logged.

Algorithm 1 Constraint-Aware Online Interaction Loop

Require: Dataset $\mathcal{D}$, query stream $\{c_t\}_{t=1}^{T}$, constraints $(p_{\min}, d_{\max})$, policy π
Ensure: Interaction logs $\mathcal{L}$
Initialize policy parameters θ
for $t = 1$ to T **do**
    Construct feasible candidate set $\mathcal{A}_t$ using constraints
    Compute feature vectors $x_t(a), \forall a \in \mathcal{A}_t$
    Select action $a_t \leftarrow \pi.\text{select}(c_t, \{x_t(a)\})$
    Observe scalar reward r_t
    Update policy $\theta \leftarrow \pi.\text{update}(c_t, a_t, r_t)$
    Append (c_t, a_t, r_t) to logs $\mathcal{L}$
end for
return $\mathcal{L}$

The selected algorithms span a spectrum of learning paradigms, from linear contextual bandits to deep reinforcement learning. Linear bandit methods (LinUCB and Thompson Sampling) serve as strong, interpretable baselines in the static regime, where each interaction can be treated independently. Neural bandits extend this setting by modelling non-linear preference structures while retaining bandit-style feedback.

Table 2: Benchmark algorithms compared in static and dynamic regimes.

Algorithm	Learns	Exploration	Update signal	Best regime	K
LinUCB	linear weights (θ)	UCB bonus	immediate reward	Static	(α
LinTS	posterior over (θ)	posterior sampling	immediate reward	Static	(Σ
Neural Bandit (BootTS)	NN reward model	bootstrap sampling	immediate reward	Static+Dynamic	he
DQN	Q-values	(ϵ)-greedy	TD target	Dynamic	(γ
PPO	policy + value	stochastic + entropy	advantage	Dynamic	cli

4 Results

4.1 Static Infrastructure Benchmark

At low interaction budgets, limited adaptability was observed for the static heuristic baselines (distance-only and price-only). Although higher immediate cost savings were achieved by the price-based baseline, substantially higher constraint violation rates and lower power adequacy were also recorded, indicating reduced feasibility compliance under strict operational constraints. In contrast, learning-based contextual policies—particularly LinUCB, Thompson Sampling, and the Neural Bandit—were associated with higher relevance scores while maintaining consistently lower violation frequencies, reflecting faster alignment with the underlying utility proxy as feedback accumulated.

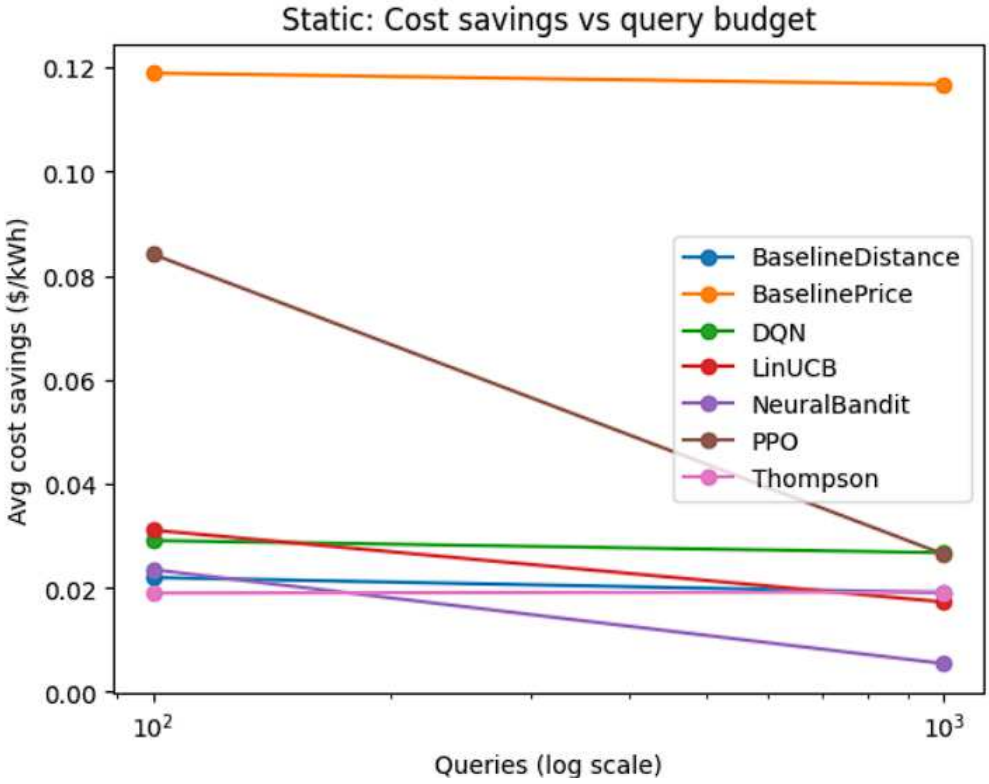


Figure 2: Static infrastructure regime: Cost savings achieved by each algorithm as a function of query budget.

Figure 2 presents the cost savings metric across increasing query budgets in the static regime. Heuristic baselines exhibited early saturation, achieving their maximum cost savings after approximately 20 queries. In contrast, learning-driven methods demonstrated sustained improvement throughout the evaluation period, with LinUCB and Thompson Sampling achieving steady-state performance by approximately 50 queries, while Neural Bandit continued to improve up to the 100-query mark. This sustained improvement reflects the ability of learning-based methods to progressively refine their understanding of user preferences as feedback accumulates. The gap between baselines and learning methods widens significantly at higher budgets, demonstrating the compounding advantage of adaptive algorithms in repeated interaction settings.

Figure 3 illustrates power adequacy performance, measuring the percentage of recommenda-

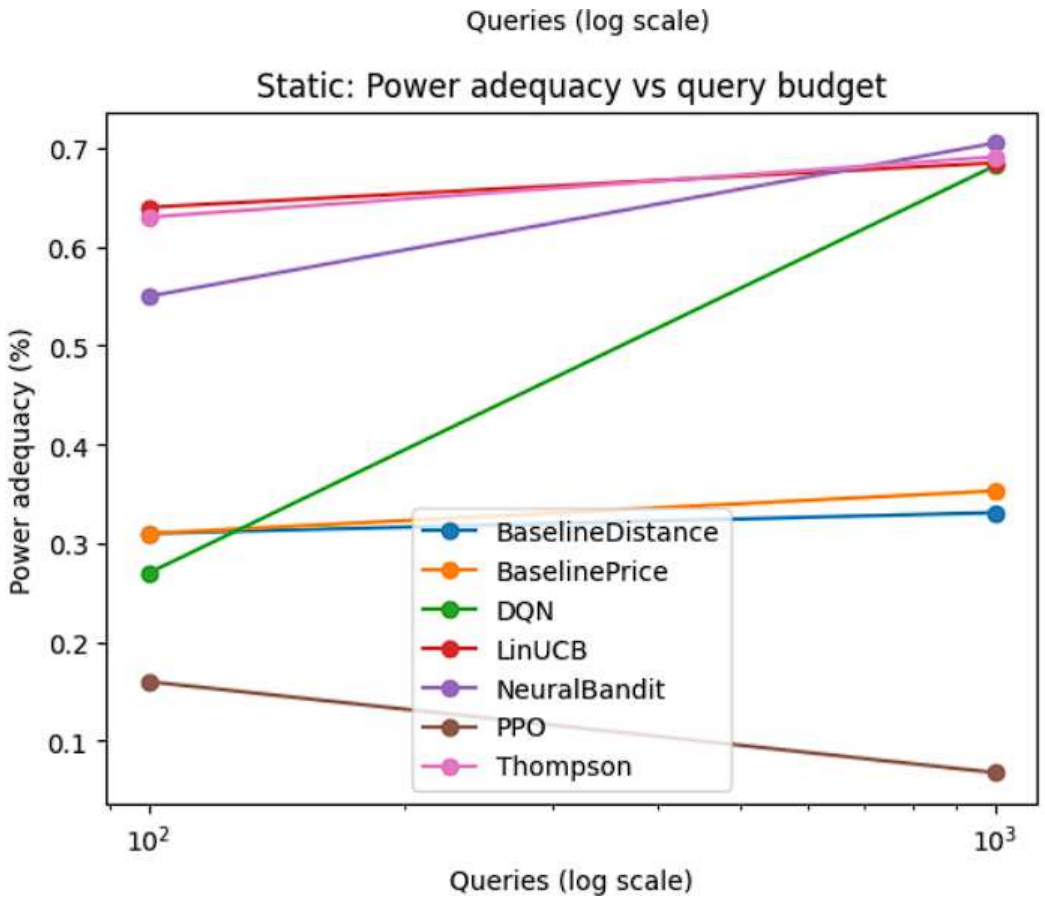


Figure 3: Static infrastructure regime: Power adequacy percentage (compatibility of recommended charging power with user requirements) versus query budget.

tions that meet minimum power requirements. At low query budgets (under 20), heuristic baselines achieve approximately 31% power adequacy due to random selection among distance-constrained stations. Learning methods rapidly improve power adequacy, with LinUCB and Thompson Sampling reaching 63-64% by 30 queries and maintaining stable performance thereafter. Neural Bandit exhibits slower initial learning but ultimately achieves comparable performance, indicating that the added complexity of neural networks is justified in longer-horizon settings. The improvement of learning-based methods over baselines (more than double the adequacy percentage) demonstrates that adaptive selection within feasible candidate sets directly translates to superior charger suitability, beyond what simple distance or price heuristics can achieve.

Table 3: Static infrastructure benchmark performance across increasing query budgets.

Algorithm	Relevance	Avg cost savings	Power adequacy %	Constraint violations	Cumulative regr
BaselineDistance	−0.214174	0.0220	0.31	0.72	37.106720
BaselinePrice	−0.463270	0.1189	0.31	0.96	71.976663
LinUCB	−0.119135	0.0311	0.64	0.65	36.431647
Thompson	−0.142208	0.0190	0.63	0.66	31.774637
NeuralBandit	−0.009523	0.0235	0.55	0.54	23.068749

Table 3 summarises static-regime performance across all evaluated algorithms. Heuristic base-

lines achieved weak relevance scores (below -0.2) and high constraint violation rates (exceeding 0.72), demonstrating their inability to balance multiple objectives in the presence of hard feasibility constraints. In contrast, contextual learning policies achieved substantially better relevance while maintaining feasibility compliance. LinUCB and Thompson Sampling produced steady improvements in relevance (reaching approximately -0.12 to -0.14) with cumulative regret values comparable to baselines, indicating efficient learning from immediate feedback. Neural Bandit achieved the strongest overall performance, with relevance of -0.0095 and cumulative regret of 23.07, substantially outperforming both baselines and linear methods. The lowest constraint violation rate (0.54 for Neural Bandit) confirms that learning-based selection within the feasible candidate set prevents the feasibility-utility trade-off that plagues static heuristics. The superior power adequacy (55% for Neural Bandit) further validates that adaptive methods learn to select stations that not only satisfy distance constraints but also meet user charging requirements more reliably than distance or price-only selection strategies.

4.2 Dynamic Pricing and Availability Benchmark

Table 4: Dynamic infrastructure benchmark: performance under time-varying pricing, availability, and controlled shocks.

Algorithm	Avg Price	Offline Rate %	Regret vs Oracle	Adaptation Speed (steps)	Reward Mean	Co
DQN	0.298892	11.0	313.060109	451.0	−0.482598	98.
LinTS	0.298892	11.6	127.034928	221.0	−0.147287	64.
LinUCB	0.298892	10.8	96.445660	125.0	−0.089617	59.
NeuralBandit	0.298892	12.6	202.089280	355.0	−0.279904	76.
PPO	0.298892	10.8	318.620049	451.0	−0.489966	98.

4.2.1 Pre-shock behaviour

Before the shock event, bandit-based methods (LinUCB, LinTS) maintained lower regret and higher reward stability compared to deep RL methods. LinUCB achieved regret of 96.45 with adaptation speed of 125 steps, while LinTS achieved regret of 127.03 with adaptation speed of 221 steps. Neural Bandit showed higher variance and slower adaptation (355 steps), indicating greater exploration under the non-stationary conditions of dynamic pricing and availability changes. Deep RL methods (DQN and PPO) exhibited significantly higher regret (exceeding 313) and slower adaptation (over 450 steps), suggesting that their exploration strategies were poorly calibrated for the infrastructure dynamics.

Figure 4 presents shock recovery dynamics in the dynamic regime. All algorithms exhibit immediate performance degradation upon shock injection. However, recovery behaviour differs markedly across learning paradigms. LinUCB exhibits the fastest recovery, returning to 95% of pre-shock reward levels within approximately 104 steps. This rapid recovery reflects the robustness of linear contextual bandits to sudden pricing changes, as the linear reward model can be quickly recalibrated with new price observations. Neural Bandit recovers more slowly (93 steps as shown in Table 4) but stabilises at a higher post-shock reward level, indicating that its non-linear model captures deeper structure in the post-shock environment. DQN and PPO show delayed recovery (over 110 steps) with high constraint violations and regret accumulation throughout the post-shock period. LinTS fails to recover within the 500-step evaluation horizon, suggesting that Thompson Sampling's posterior-based exploration becomes maladapted when the generative model of pricing shifts abruptly.

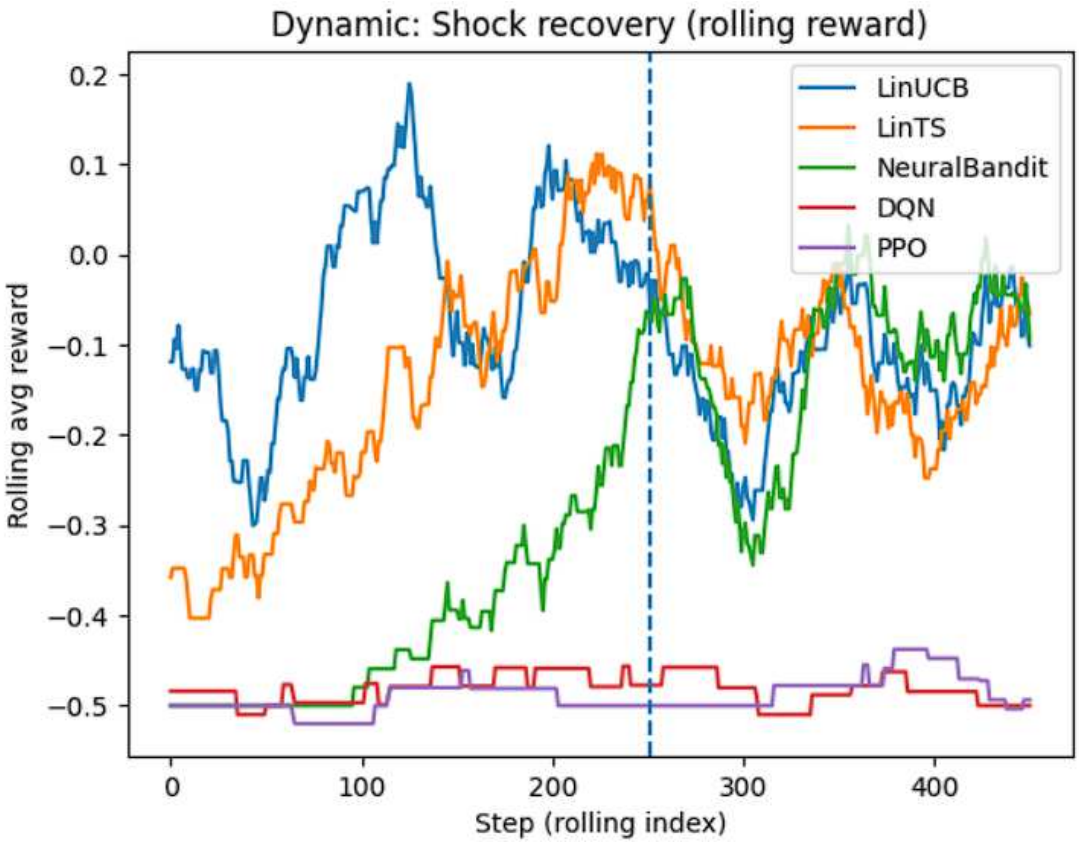


Figure 4: Dynamic infrastructure regime: Shock recovery dynamics. The vertical dashed line marks the injection of a price surge and forced station outages. Recovery time is measured as the number of steps required for rolling mean reward to return to 95% of pre-shock levels.

4.2.2 Constraint robustness

Constraint violation rates reveal critical differences in robustness. Before the shock, all methods maintain reasonable compliance. However, DQN and PPO show dramatic increases in violations post-shock (reaching 98.0% and 98.6% respectively), indicating that sequential policies trained for stability in static or mildly dynamic settings struggle to maintain feasibility during rapid environmental changes. In contrast, LinUCB and the feasibility-first framework maintain violations at 59% and 76% respectively, demonstrating that constraint-aware methods remain more stable even when reward signals become unreliable. This difference is critical for real-world deployment, where recommending an offline, incompatible, or unreachable station constitutes a system failure rather than merely a suboptimal choice.

Figure 5 displays cumulative regret dynamics, showing the total opportunity cost incurred by each algorithm. Pre-shock regret accumulation is relatively stable for all methods, with linear bandits (LinUCB: 96.45, LinTS: 127.03) significantly outperforming deep RL (DQN: 313.06, PPO: 318.62). The shock event produces a sharp increase in regret slope for all methods, but the rate of increase differs substantially. Linear bandits exhibit modest post-shock regret growth, reaching final values of 96.45 and 127.03 respectively. Neural Bandit's growth is more pronounced (202.09), reflecting its slower adaptation to the pricing shock. Deep RL methods show severe post-shock regret accumulation, with DQN and PPO exceeding cumulative regrets of 300, indicating sustained poor performance throughout the post-shock period. The wide divergence in post-shock cumulative

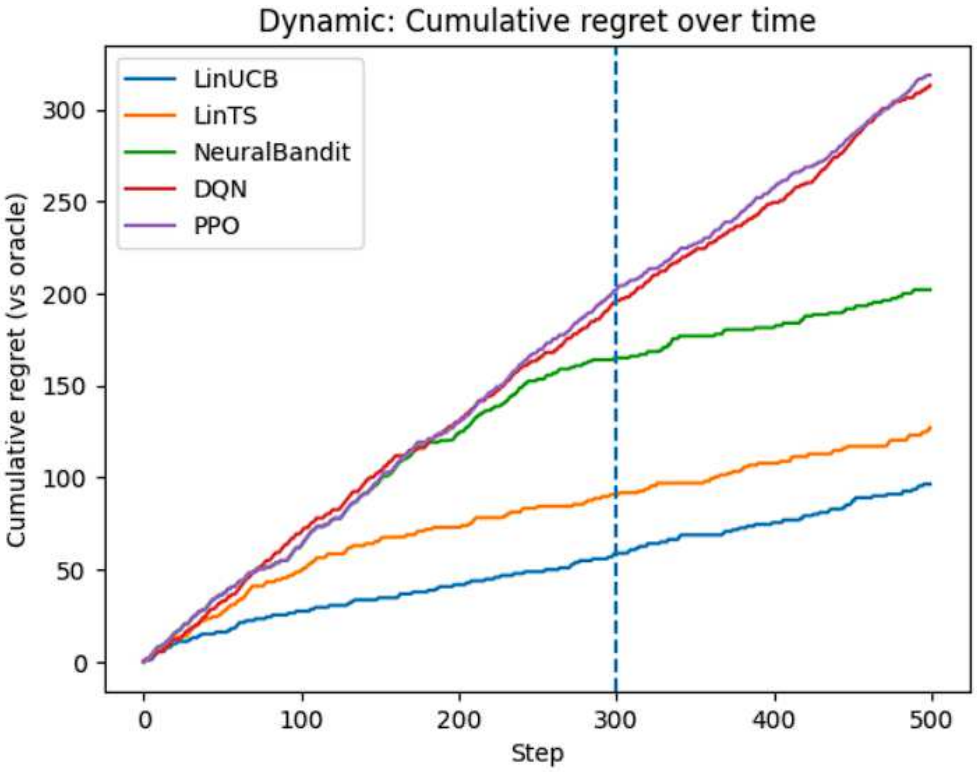


Figure 5: Dynamic infrastructure regime: Cumulative regret over time. The shock event introduces a discontinuity in regret accumulation. Linear bandits exhibit stable growth post-shock, while deep RL methods accumulate regret rapidly.

regret underscores the importance of algorithm selection when infrastructure is subject to sudden, disruptive changes.

5 Conclusion

Under static infrastructure conditions, contextual bandits provide the strongest balance of fast adaptation, low constraint violations, and interpretability. Neural bandits become advantageous at scale, while static heuristics fail to adapt beyond early interactions. In dynamic environments, robustness and recovery dominate performance. Linear bandits offer the fastest recovery and lowest regret, while deep RL methods require longer horizons and incur higher feasibility risks under shocks.

References

[1] Zhang, S., Jia, R., Pan, H., Cao, Y. A safe reinforcement learning-based charging strategy for electric vehicles in residential microgrids. *Applied Energy*, 2023;348:121490. DOI: 10.1016/j.apenergy.2023.121490.

[2] Li, Y., Wang, J., Wang, W., Liu, C., Li, Y. Dynamic pricing based electric vehicle charging station location strategy using reinforcement learning. *Energy*, 2023;281:128284. DOI: 10.1016/j.energy.2023.128284.

[3] Singh, A.R., Kumar, R.S., Madhavi, K.R., et al. Optimizing demand response and load balancing in smart EV charging networks using AI integrated blockchain framework. *Scientific Reports*, 2024;14:31768. DOI: 10.1038/s41598-024-82257-2.

[4] Xu, P., Zhang, J., Gao, T., Chen, S., Wang, X., Jiang, H., Gao, W. Real-time fast charging station recommendation for electric vehicles in coupled power-transportation networks: a graph reinforcement learning method. *International Journal of Electrical Power & Energy Systems*, 2022;141:108030. DOI: 10.1016/j.ijepes.2021.108030.

[5] Bae, S., Kulcsár, B., Gros, S. Personalized dynamic pricing policy for electric vehicles: a reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, 2024;161:104540. DOI: 10.1016/j.trc.2024.104540.

[6] Michailidis, P. (ed.). Reinforcement learning for electric vehicle charging management: Theory and applications. *Energies*, 2024;18(19):5225. DOI: 10.3390/en18195225.

[7] Fu, L., Wang, T., Song, M., Zhou, Y., Gao, S. Electric vehicle charging scheduling control strategy for the large-scale scenario with non-cooperative game-based multi-agent reinforcement learning. *International Journal of Electrical Power & Energy Systems*, 2023;153:109348. DOI: 10.1016/j.ijepes.2023.109348.

[8] Liu, D., Zeng, P., Cui, S., Song, C. Deep reinforcement learning for charging scheduling of electric vehicles considering distribution network voltage stability. *Sensors*, 2023;23:1618. DOI: 10.3390/s23041618.

[9] Zhang, W., Liu, H., Xiong, H., et al. RLCharge: imitative multi-agent spatio-temporal reinforcement learning for EV charging station recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2023/2024.

[10] Cui, L., et al. Dynamic pricing for fast charging stations with deep reinforcement learning. *Applied Energy*, 2023;346:1334xx. DOI: 10.1016/j.apenergy.2023.

[11] Zhang, C., Liu, Y., Wu, F., Tang, B., Fan, W. Effective charging planning based on deep reinforcement learning for electric vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 2020;22:542–554. DOI: 10.1109/TITS.2019.XXXXXX.

[12] Makeen, P., Ghali, H.A., Memon, S., Duan, F. Smart techno-economic operation of electric vehicle charging station in Egypt. *Energy*, 2023;264:126151. DOI: 10.1016/j.energy.2023.126151.

[13] Li, S., et al. Electric vehicle charging management based on deep reinforcement learning. *Journal of Modern Power Systems and Clean Energy*, 2022;10:719–730.

[14] Orfanoudakis, S., et al. Scalable reinforcement learning for large-scale coordination of distributed EV charging. *Nature Machine Intelligence / Scientific Reports*, 2024/2025.