Methodology

Data Preparation

The methodology relied on four structured data components: tariffs, charging stations, vehicle specifications with charging curves, and simulated user demands.

**Tariff data** were collected directly from operator websites (SureCharge, Believ etc.). The raw data exhibited heterogeneous formats, including per-kWh energy rates, session fees, idle charges, and membership discounts etc.

A screenshot of a computer

AI-generated content may be incorrect.A close-up of a message

AI-generated content may be incorrect.

Figure 1 Raw Tariff Information Examples

To enable consistent cost computation, we converted all rates to £/kWh and decomposed the tariffs into three structured datasets. ((i) base tariffs, (ii) conditional rules (e.g., idle or time-of-use fees), and (iii) connector-specific overrides). This ensured realistic and provider-specific cost estimation in the simulation environment.

A diagram of a diagram

AI-generated content may be incorrect.

Figure 2 Conversion Process

**Charging station data** were obtained from the UK National ChargePoint Registry (NCR) {CITATION NEEDED}. After removing duplicates and fixing coordinate errors, we split the dataset into two layers: station-level metadata (for spatial queries) and connector-level features (for action feasibility in the RL environment).

A diagram of a system

AI-generated content may be incorrect.

Figure 3 Splitting the Charging Data

**Vehicle specifications** were sourced from Open EV Data v2 {CITATION NEEDED}. Charging curves were interpolated over 0–100% state of charge (SoC), expressed in kWh/km, and capped by station power limits to model charging times and SoC changes.

A diagram of a diagram

AI-generated content may be incorrect.

Figure 4 Splitting the EV Data

**Simulated users** were generated as structured trip instances combining user profiles, EV states, trip definitions, and behavioral constraints (see Table 1).

|  |  |
| --- | --- |
| **Feature Category** | **Examples** |
| User Profile | Membership type (Member/Payg), Subscription status, Sessions per month |
| Trip Definition | Origin-Destination coordinates, Departure time, Trip distance |
| EV State | Assigned EV model, Start SoC, Reserve SoC, Available energy |
| Preferences & Constraints | Optimisation objective, max detour km |
| Energy Efficiency | Consumption in kWh/km, estimated required kWh per trip |

Table 1 Feature Categories & Examples

Together, these four components provided a unified and reproducible dataset for training and evaluating the RL agents. By integrating these, the environment captures both system-level constraints and realistic user behaviour.

**Environment Design**

The charging decision-making task was formalised as a Markov Decision Process (MDP) {CITATION NEEDED}, defined as

where S is the state space, A the action space, P the transition dynamics, R the reward function, and the discount factor.

**State vector,** Observations include SoC, vehicle position, distance to destination, and attributes of up to k candidate stations (connector type, power, tariff, detour cost).

**Action space**. The agent selects one of the top-k stations. If fewer are available, remaining slots are padded with dummy actions that incur penalties if chosen.

**Transition dynamics**. Driving reduces SoC and advances time (with traffic effects), while charging replenishes SoC according to nonlinear curves capped by connector power, with a fixed three-minute overhead per session.

**Reward function.** Supports cost, time, and hybrid objectives, with shaping terms: +50 for success, –200 for depletion, –2 for infeasible choices (see Section 3.3).

**Traffic modelling**. The environment offers three modes: constant-speed fallback (25 km/h), lightweight congestion multipliers (×1.6 AM, ×1.5 PM, ×1.0 off-peak), and SUMO-backed microscopic simulation {CITATION NEEDED}. While multipliers were useful for prototyping, all reported experiments employed SUMO for realistic congestion and routing.

**Design contributions.** To ensure non-trivial learning, initial SoC was sampled low (10–30%) and trip lengths set to 12–25 km, even though many real London trips would not require charging. Each charging session included a three-minute overhead, and infeasible actions triggered penalties. These adjustments deviate from typical travel but were necessary to prevent trivial episodes and provide sufficient learning signals, reflecting a trade-off between realism and trainability.

Together, these design elements produced an RL environment that was both computationally feasible and sufficiently realistic to evaluate charging strategies under urban driving conditions. While simplified in some respects, the use of SUMO-backed traffic and structured user demands ensured that the environment captured the essential challenges of EV route planning in Inner London.

**Reward Design**

The environment supported three primary reward definitions:

A screenshot of a computer

AI-generated content may be incorrect.

where TtT\_tTt​ is total journey time in minutes and CtC\_tCt​ is cost in GBP. The denominators normalise magnitudes to prevent either factor from dominating.

**Reward shaping** was applied to provide stronger learning signals. Successful trip completion yielded a bonus of +50, while battery depletion triggered a –200 penalty, set deliberately high to enforce feasibility. Selecting an infeasible charging station produced a –2 penalty, discouraging wasted actions.

**Indirect influences.** Several environment factors indirectly shaped the reward:

* A three-minute overhead per charging session discouraged both micro-charging and excessive station hopping.
* Traffic conditions (constant speed, multipliers, or SUMO congestion) embedded realistic delay costs.
* Charging efficiency (92%92\%92%) and connector power limits affected how much usable SoC could be gained, making some cheap stations unattractive.
* Reserve SoC constraints penalised borderline trips by reducing feasibility.

**Hybrid scaling.** A value of time of £0.05/minute {CITATION NEEDED} was used to align cost and time on a consistent scale, preventing degenerate policies that optimised one metric while ignoring the other (see Fig. 4).

Together, these elements produced a reward landscape that captured realistic cost–time trade-offs while remaining learnable.

Proximal Policy Optimization

Training employed **Proximal Policy Optimisation (PPO)** {CITATION NEEDED}, implemented via the Stable-Baselines3 library {CITATION NEEDED}. No modifications were made to the algorithm itself.

The clipped surrogate objective {CITATION NEEDED} is defined as:

A black screen with white text

AI-generated content may be incorrect.

While PPO provided the policy optimisation framework, our contribution lay in **the custom environment, reward shaping, and evaluation design**, rather than in modifications to the algorithm. The PPO training loop is shown in Fig. 5