FROM REAL ROADS TO DEPOT-SCALE ROUTING: A DATASET AND BENCHMARK FOR ELECTRIC VEHICLE ROUTING WITH TIME WINDOWS

Anonymous authors
Paper under double-blind review

ABSTRACT

Solvers for the Electric Vehicle Routing Problem with Time Windows (EVRP-TW) are commonly evaluated on small synthetic benchmarks. Whether conclusions drawn from such benchmarks generalize to operational workloads on real road networks remains unclear. We introduce EVRP-TW-D/B, a public dataset-and-benchmark suite. EVRP-TW-D provides 71,500 instance views across 11 U.S. cities and 100–2,000 customers. Its two-stage generator builds reusable city environments from directed road networks and instantiates delivery days using joint demand, service-time, and time-window tuples from Amazon records. EVRP-TW-B compares five learning-based policies with exact and metaheuristic baselines using a common task and objective, released implementations, independent verification, and declared budgets. With 100-customer Euclidean training, only two policies achieve lower mean costs than the best evaluated search baseline under a 30-minute budget at 500 and 1,000 customers. Training the same architectures on 100-customer EVRP-TW-D instances yields full feasibility and lower mean costs than this baseline at both scales. Policies trained at 1,000 customers retain this advantage beyond the training geography. These findings motivate evaluating solver competitiveness across operational workloads and geographies, with EVRP-TW-D/B providing a reproducible foundation.

1 INTRODUCTION

Last-mile delivery is going electric (Basma et al., 2022; Patella et al., 2020), and that changes what a route is. In conventional fleets, refueling is infrequent and fast enough to sit outside the plan, so routing reduces to sequencing customers under cargo and time-window constraints (Laporte, 2007; Toth & Vigo, 2002). A battery-electric fleet also faces an energy constraint. Fixed vehicle-deployment costs encourage consolidating deliveries into fewer routes, potentially increasing the distance each vehicle travels. Whether en-route charging is needed therefore depends on route assignment as well as battery capacity. When needed, charging must be scheduled alongside customer visits because it consumes time within the delivery horizon; routes that fit within one charge require no charging stops. The *Electric Vehicle Routing Problem with Time Windows* (EVRP-TW) (Schneider et al., 2014) formalizes this trade-off, planning customer visits and charging jointly under cargo, battery, and time-window constraints.

Exact optimization, metaheuristics, and deep reinforcement learning (DRL) have all been developed for the EVRP-TW (Xia et al., 2024; Moradi et al., 2024; Duan et al., 2023; Bi et al., 2023). Representative classical benchmarks use synthetic Euclidean instances with up to 100 customers (Table 1). However, performance on these small, stylized instances may not generalize to real-world delivery operations, where problem scale, road-network structure, and operational constraints can be substantially different. This motivates a central question:

> *Does strong performance on small synthetic benchmarks still hold at operational scale on real road networks?*

Answering this question requires an evaluation foundation that addresses three disparities between classical benchmarks and operational delivery conditions.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

D1: problem and dataset scale. Scale has two dimensions: customers per instance N, and the number of available instances M. Larger N is needed to evaluate solver behavior as delivery workloads grow. Sufficient M is needed to evaluate variation across delivery realizations and, for learned solvers, to provide separate training and held-out data. Classical Solomon-derived benchmarks provide reproducible reference cases, including 56 instances at 100 customers (Solomon, 1987; Schneider et al., 2014), but a fixed collection does not support generating further workloads and delivery realizations under a documented procedure.

D2: grounding of delivery tasks. Demand, service time, and time windows jointly determine which customers can share a feasible route. Their generation therefore shapes the optimization problem itself. Sampling them without an explicit empirical reference leaves unclear which operational delivery conditions the resulting instances represent.

D3: representation of travel. Customer coordinates do not determine road connectivity, direction, or detours. A geometrically nearby charging station can require a long detour, increasing energy use and reducing the time available for later deliveries. Euclidean evaluation omits road-network effects on travel distances and times (Son et al., 2026), which can affect route cost and feasibility.

To address these disparities, we introduce EVRP-TW-D, a public EVRP-TW corpus combining depot-scale workloads, directed road networks, and operational delivery data. Its two-stage generator builds reusable *City Logistics Environments* (CLEs) from public geographic and infrastructure data, then instantiates delivery days. Using depot-day workloads as the reference for N, with a median of 1,388 usable drop-off stops across 521 Amazon source station-days (Merchán et al., 2024), the corpus spans 100–2,000 customers. It provides 71,500 main benchmark views, while the generator supports further realizations under documented sampling and partitioning rules, addressing M (D1). Demand, planned service time, and time windows are transferred as joint tuples from Amazon data (D2). Travel times and distances follow fastest-time paths on released directed roads, with energy derived from distance (D3).

Comparisons are further complicated by differing objectives and constraints, implementations that are difficult to reproduce without code and configurations (Tang et al., 2025), and inconsistent budgets, aggregation cohorts, and quality references. EVRP-TW-B therefore fixes a shared task and operating-cost objective, releases baseline implementations and configurations, independently verifies feasibility and recomputes cost, and reports feasibility, solution quality, and inference or search time under declared budgets, explicit cohorts, and a disclosed reference pool. Under these budgets, small-scale Euclidean training does not ensure competitive performance at operational scale, whereas policies trained on EVRP-TW-D can outperform the strongest evaluated search baseline and transfer geographically (Section 6).

Our contributions are threefold. **(1) A public corpus at operational scale** spanning 11 U.S. cities, with directed-road travel and operationally grounded delivery attributes. **(2) A reusable two-stage generator** with predefined splits for new delivery days, held-out communities, and an unseen city. **(3) A cross-paradigm benchmark** with released implementations and run records for reproducible comparisons across exact, metaheuristic, and learning-based solvers.

2 Related Work

Datasets and generators for EVRP-TW. Classical EVRP-TW benchmarks extend Solomon's VRPTW instances with battery and charging parameters (Solomon, 1987; Schneider et al., 2014), retaining fixed collections with up to 100 customers per instance, synthetic delivery attributes, and Euclidean travel. Recent EVRP-TW and related routing resources advance three complementary capabilities. *Configurable generation* includes SynthCharge, with feasibility screening (Daysalilar et al., 2026), and SVRPBench's stochastic VRP generator, with congestion, delays, and empirically motivated time-window models over Euclidean travel (Heakl et al., 2025). *Road-derived travel* appears in time-dependent benchmarks built from OpenStreetMap and Uber Movement (Blauth et al., 2024), EVRPGen's OSM-based generation (Akbay & Blum, 2025), and RRNCO's asymmetric distance and duration matrices for 100 cities with predefined splits (Son et al., 2026); these generation schemes use prescribed or synthetic delivery requirements. *Observed delivery data* are supplied by Amazon Last Mile records (Merchán et al., 2024), whose released task is sequencing stops within preassigned single-vehicle routes, leaving fleet sizing, customer assignment, and charging outside

Table 1: Public EVRP-TW instance collections and generators. M counts released instances; n/a: generator without a fixed instance set.

Resource	Max N	Instances M	Released generator	Travel representation	Recorded delivery attrs.
Schneider et al. (2014)	100	92	✗	Euclidean	✗
Desaulniers et al. (2016)	100	168	✗	Euclidean	✗
Lam et al. (2022)	100	448	✗	Euclidean	✗
Rastani et al. (2026)	50	124	✗	Euclidean	✗
SynthCharge (2026)	n/a	n/a	✓	Euclidean	✗
Goeke–Schneider (2015)	100	180	✗	Road-based matrices	✗
Goeke (2019)	100	92	✗	Euclidean	✗
Bruglieri et al. (2025) (benchmark)	100	30	✗	Euclidean	✗
Bruglieri et al. (2025) (real case)	30	2	✗	Road-based matrices	✗
Peng et al. (2025) (benchmark)	100	60	✗	Euclidean	✗
Peng et al. (2025) (real case)	47	1	✗	Distance/time matrices	✓
EVRPGen (2025)	n/a	n/a	✓	Directed roads	✗
EVRP-TW-D (Ours)	**2,000**	**71,500**	✓	Directed roads	✓

its scope. The missing combination among these resources is reusable EVRP-TW generation with depot-scale workloads, travel on released directed road networks, and joint delivery-attribute tuples from operational records (Table 1). EVRP-TW-D provides this combination through its two-stage generator, with predefined splits for new delivery days, held-out communities, and an unseen city.

Solvers and conditions for comparison. Exact methods use mixed-integer or route-based formulations (Desaulniers et al., 2016; Cataldo-Díaz et al., 2024; Nafstad et al., 2025), while metaheuristics combine VNS/TS or ALNS with charging-insertion operators (Schneider et al., 2014; Keskin & Çatay, 2016; Duan et al., 2023); both optimize each instance without a pretrained policy. Learned approaches include attention-based construction and multi-start training (Kool et al., 2018; Kwon et al., 2020); their policies additionally depend on the training environment: Lin et al. (2021) and Tang et al. (2025) train on Euclidean travel. The representation of travel also varies: Chen et al. (2022), Edge-DIRECT for heterogeneous fleets (Mozhdehi et al., 2024), and RRNCO (Son et al., 2026) explicitly encode pairwise travel quantities.

Across these families, task definitions vary in charging policy, fleet composition, charging-time model, and objective (Appendix G, Table 25(a)). Evaluation budgets include fixed decoding sample counts (Mozhdehi et al., 2024) and one-hour time limits (Lam et al., 2022). Reporting rules differ as well: the latter study counts both proven optima and proofs of infeasibility as solved, which is distinct from counting instances with a feasible delivery plan. Implementation access further affects traceability: TERRAN reports reconstructing VNS/TS because the original implementation was unavailable (Tang et al., 2025) (Table 25(b)). Cross-paper results alone therefore cannot establish how these solvers compare on a common road-based task. EVRP-TW-B provides a shared task and objective, released reference implementations, independent verification, and common reporting rules under declared budgets. RRNCO already compares noise-perturbed and road-derived training data within its model (Son et al., 2026); within our benchmark, we compare Euclidean- and road-trained versions of EVRP-TW policies under coupled battery, charging, and time-window constraints.

3 EVRP-TW Problem Setting

Task and assumptions. An EVRP-TW instance specifies a depot, customers with volume demands, service durations, and hard time windows, and charging stations. The task is to serve every customer exactly once using electric vehicles, jointly determining fleet size, customer assignment, and the sequence of customer visits and charging stops on each route. Appendix A provides mixed-integer and Markov decision process formulations of this reference task for exact optimization and constructive learning, respectively.

Following Schneider et al. (2014), we assume homogeneous vehicles, single-trip operation, distance-proportional energy consumption, and full recharging, with no hard fleet-size limit. These assumptions define a common reference task for assessing the strengths and limitations of different solver families under shared constraints. Holding them fixed lets us test whether established methods remain effective on larger, road-based workloads within the classical task itself, without adding vehicle-type selection, multi-trip scheduling, or charging-amount decisions. Extensions such as partial recharging or load-dependent energy require an explicit statement of which reference assumptions they relax (Appendix D.6).

Each deployed vehicle completes one depot-to-depot route, so vehicle and route counts coincide. All goods for a route are loaded at departure, and their total volume demand cannot exceed vehicle cargo capacity. Service takes the specified duration and must begin within each customer's time window; early arrivals may wait. Travel from i to j follows a precomputed fastest-time road path, whose distance and travel time are denoted by d_{ij}^{rt} and t_{ij}^{rt}, respectively (Section 4). These quantities are static within an instance and may be asymmetric. Each vehicle starts with a full usable battery and consumes $e_{ij} = \kappa d_{ij}^{\mathrm{rt}}$ on each path, where κ is a constant consumption rate per kilometer. Battery energy must remain nonnegative throughout the route. Each charging visit restores full usable capacity; charging time is the replenished energy divided by the station's constant effective power. All vehicles may use every station in the instance without queuing, and stations may be revisited by the same or different vehicles. Time is measured from the start of the operating horizon, including customer-window endpoints. Each deployed vehicle departs at time 0 and must return by $T_{\max}$. The depot and all stations are available throughout $[0, T_{\max}]$; travel, waiting, service, and charging share this horizon, so recharging extends range at the cost of slack for later deliveries.

Reference objective. Daily operating cost captures the trade-off between fleet deployment and driving energy. For a feasible plan x, we minimize $C(x) = F_{\mathrm{day}} K(x) + p_e \kappa D(x)$, where $K(x)$ counts deployed vehicles and $D(x)$ sums distance along the specified fastest-time paths, including charging detours and depot returns. The fixed daily charge $F_{\mathrm{day}} = F_{\mathrm{veh}} + F_{\mathrm{drv}}$ covers vehicle and driver costs per deployed vehicle; the travel term prices consumed energy $\kappa D(x)$ at a uniform electricity price p_e. The reference calibration is specified in Section 5.

4 EVRP-TW-D: A Road-Grounded EVRP-TW Corpus

EVRP-TW-D comprises a released corpus and a two-stage generator (Figure 1). Roads and facilities persist across delivery days, while active customers and their requirements change. Stage 1 therefore builds a reusable *City Logistics Environment* (CLE), and Stage 2 instantiates delivery days within it. Reusing this city layer lets workload sizes and delivery realizations vary without rebuilding the geographic setting.

4.1 The Two-Stage Instance Generator

Stage 1: City Logistics Environments. A CLE combines a directed road network, reference speed profiles, and candidate service and facility locations. Given a city boundary and coverage by the public sources below, the released generator constructs a CLE; the corpus is one instantiation of this process.

Road network. Land-only Census Place boundaries delimit location selection (U.S. Census Bureau, 2025); OpenStreetMap (OSM) supplies road geometry, lengths, and one-way restrictions (OpenStreetMap contributors, n.d.). Connecting roads outside the boundary are retained, allowing routes to leave and re-enter the city. Candidate locations must admit directed round trips (Appendix C.2).

Reference speed profiles. Road lengths alone do not determine travel times. OSM speed limits, supplemented by the Highway Performance Monitoring System (HPMS) and within-city road-class fallbacks, provide a free-flow proxy (Federal Highway Administration, 2026). Relative speed factors from Motor Vehicle Emission Simulator version 5 (MOVES5) profiles are matched by delivery-vehicle type, road class, and weekday or weekend type (U.S. Environmental Protection Agency, 2024). Multiplying the proxy by these factors gives reference speeds; road lengths divided by these speeds determine edge travel times and hence fastest-time paths (Appendix C.3).

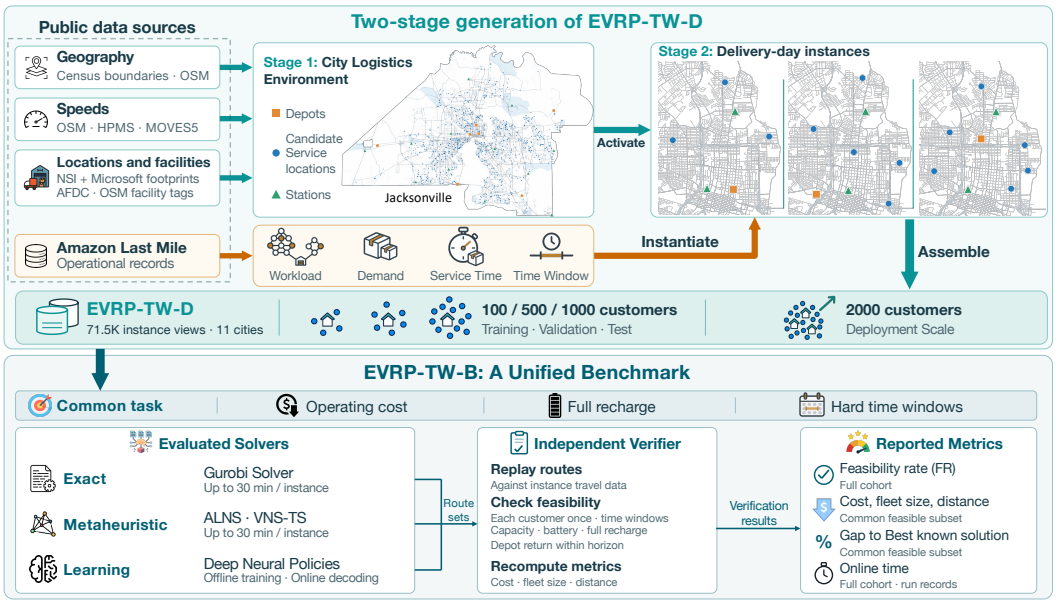


Figure 1: **Overview of EVRP-TW-D/B.** *Top:* A two-stage generator produces a multiscale, road-grounded delivery-day corpus within reusable city environments. *Bottom:* Solver families are compared through a common task, independent verification, and shared reporting rules.

Service and facility locations. Residential service candidates require both use evidence and building geometry. We match National Structure Inventory (NSI) residential-use records to Microsoft building footprints to obtain both (U.S. Army Corps of Engineers, 2026; Microsoft Maps, 2018). OSM warehouse, logistics, and dispatch evidence supplies depot candidates. Publicly available Alternative Fuels Data Center (AFDC) sites within the boundary supply charging candidates (National Laboratory of the Rockies, n.d.). Each location attaches to its nearest eligible road (Appendix C.4).

Stage 2: Delivery-Day Instances. Stage 2 constructs a *base instance* containing N_{target} customers. Nested partitioning of its customer pool defines the released *instance views*, which together form an *instance family* (Section 4.2).

Workload structure and customer placement. An eligible depot is paired with a compatible *structure template* from Amazon Last Mile records (Merchán et al., 2024). The template specifies customer counts across source route groups and depot-to-stop travel-time bands, guiding depot-level demand placement without prescribing vehicle assignments or routes. To realize these counts spatially, we group service candidates into *communities* using Census Block Groups and directed-road connectivity. For each template group, a *delivery region* expands from a seed through road-adjacent communities until the required bands contain enough candidates. Sampling to the template counts yields N_{target} distinct customers. All coordinates come from the CLE; no Amazon coordinates are transferred (Appendix C.6).

Charging infrastructure. Charging stations are selected after customers because their insertion detours depend on the day's depot and customer locations. Candidates are first filtered for battery-feasible round trips from the depot under the reference vehicle parameters, allowing intermediate charging at other candidate stations (Appendix C.6). A fixed-size roster is built greedily by minimizing a detour score combining the mean, 90th-percentile, and maximum insertion-detour time. A station's detour for a customer is the smaller additional road time from inserting it on either leg of the depot–customer–depot trip.

Travel quantities, delivery requirements, and feasibility. For each ordered pair of instance locations, travel time and distance follow the same fastest-time road path, with energy derived from that distance. Each customer represents one service stop, possibly aggregating several packages. Amazon *order templates* retain a source stop's package count, volume, planned service time, and time window as a joint tuple. Each customer receives a whole tuple subject to capacity and time-window feasibility. This preserves observed attribute combinations, although assignment filtering may change their distribution across the generated corpus. For each designated view, we verify that each customer admits a feasible single-customer route under its assigned tuple, the view's travel

quantities and station roster, and the constraints of Section 3. With no hard fleet-size limit or charging queues, these routes jointly certify that a feasible solution exists without characterizing the difficulty of finding a low-cost plan. A *family* is retained only if all its designated views pass this check (Appendix C.6).

4.2 The Released Corpus

The release counts *instance views*: views in a family share a city, depot, delivery day, and the delivery attributes and travel quantities of shared locations. EVRP-TW-D provides 71,500 main benchmark views from 7,500 families across eleven U.S. cities, at 100, 500, 1,000, and 2,000 customers, plus 500 paired Cus1000 control views. Archived Cus50 views are excluded from the main benchmark. Cus500 and larger views inherit their family's full roster of 50 charging stations; Cus100 greedily reselects 20 for its own customers, following the classical benchmark's station count (Schneider et al., 2014). Comparisons involving Cus100 therefore vary the complete instance configuration, not customer count alone.

Ten cities supply training and within-city evaluation; Jacksonville is reserved for geographic transfer. Whole communities and source station-days are partitioned before generation. **Training and validation.** Cus100, Cus500, and Cus1000 provide 50,000, 10,000, and 5,000 training views, respectively—five million customer occurrences per scale—plus 500 validation views each. **Evaluation at training scales.** Each scale provides 500 views per track: T1 uses new delivery days in training communities, T2 uses held-out communities in training cities, and T3 uses Jacksonville, totaling 4,500 test views. **Unseen-scale deployment.** Another 500 Cus2000 views use training communities in the ten training cities, with no policy trained at this scale. Each is paired with a Cus1000 control containing 1,000 of its customers from the same family. Within each city and split, families target a 5:2 weekday/weekend ratio. Views use day-type-matched speed profiles and source templates, with stable city, family, and source-record identifiers (Appendix D.2).

5 EVRP-TW-B: A Unified Benchmark

EVRP-TW-B addresses the three comparison obstacles identified in Section 1 across exact, metaheuristic, and learning-based solvers. For reproducibility, we release reference code and configurations for every evaluated implementation, together with per-instance routes and metrics, reference-pool identities, budgets, timing boundaries, and hardware records. Required task adaptations are distinguished from performance enhancements (Appendix E). The evaluation layer addresses task alignment through (i) and (ii), and evaluation protocols through (iii) (Figure 1, bottom). **(i) One task, one objective.** All methods use the constraints and operating-cost objective of Section 3, sharing charging policies and fleet assumptions. The objective's form is fixed, while its coefficients can be calibrated to local vehicle, driver, and electricity prices. All comparisons reported here use one reference calibration, identical for all methods. **(ii) Verification outside the solver.** An independent verifier replays submitted route sets against instance data to determine feasibility and recompute cost, vehicle count K, and distance D. Acceptance thus does not depend on a solver's constraint handling or self-reported feasibility and objective values. **(iii) One evaluation protocol.** A configuration m specifies the solver and its settings, including training corpus, training scale, and decoding settings for learned policies. For instance i in evaluation cohort $\mathcal{I}$, $x_{m,i}$ denotes the lowest-cost verified-feasible candidate produced and submitted within the declared search or decoding budget, and $x_{m,i}$ is undefined otherwise. All quality metrics use this same retained solution.

Feasibility and computation. Feasibility rate (FR) is the fraction of $\mathcal{I}$ for which $x_{m,i}$ exists. Mean online execution time also uses the full cohort, including failures. Timing measures actual wall-clock time spent generating candidates, excluding instance loading and verification; for learned policies, it covers the complete decoding workload. Offline training is reported separately.

Solution quality. We average operating cost, per-instance Gap, K, and distance D over the common feasible subset $\mathcal{I}_{\mathrm{cf}}$ of instances with accepted plans from every participating configuration, report $n = |\mathcal{I}_{\mathrm{cf}}|$, and interpret these means alongside full-cohort FR. Under our reference calibration, operating cost is $C(x) \simeq 413.63\,K(x) + 0.151751\,D(x)$ in U.S. dollars with D in kilometers; coefficients are rounded for display, and calibration sources and the unrounded values used for both optimization and scoring appear in Appendix B. Under this calibration, one additional vehicle costs

as much as the electricity for approximately $2{,}700\,\mathrm{km}$ of driving; other prices change this ratio. Reporting K and D separates fleet consolidation from shorter travel and permits repricing the same plans under other coefficients, although the plans are not reoptimized for them. Gap compares each retained plan with the best-known solution (BKS), $\mathrm{Gap}_{m,i} = 100\,(C(x_{m,i}) - C_i^{\mathrm{BKS}})/C_i^{\mathrm{BKS}}$, where C_i^{BKS} is the lowest cost for instance i in the reference pool, frozen for the reported comparisons under this calibration (Appendix E). The pool contains outputs from declared solver configurations, excluding routes used only to certify generated instances. Zero Gap denotes attainment of this reference, not proven optimality; a negative Gap denotes an improvement on it.

6 Experimental Results

The experiments center on learning-based policies, evaluating their deployment relative to established optimization methods and their sensitivity to training data, workload scale, and geography. We evaluate implementations of five learning-based baselines: AM-EVRPTW, EVRPTW-RL, DRL-TS, TERRAN, and RRNCO, with adaptations documented in Appendix E.1. Under the declared budgets, we ask: **Q1:** Do policies trained on 100-customer Euclidean instances remain feasible and competitive with exact and metaheuristic solvers on road-network instances with 100–1,000 customers, without retraining (Section 6.1)? **Q2:** Within each architecture, how do road-based training and training scale affect performance across deployment scales, including the unseen Cus2000 scale (Section 6.2)? **Q3:** Do fixed road-trained policies transfer to held-out communities and an unseen city (Section 6.3)? Finally, we measure the cost reduction obtained by additional search initialized from each instance's best solution among the five Road-1000 configurations (Section 6.4).

Data. Each evaluation cohort contains 500 instances. Unless stated otherwise, Cus100–Cus1000 use T1, Cus2000 uses the unseen-scale cohort in training cities, and geographic transfer uses T2/T3 (Section 4.2). Euclid-n and Road-n denote training on n-customer Euclidean instances generated as in Tang et al. (2025) and on EVRP-TW-D training views, respectively. **Solvers.** Gurobi, using the formulation in Appendix A, ALNS, and VNS/TS serve as optimization baselines. Each receives up to 1,800 seconds per instance, a budget reflecting same-day dispatch planning in fast-changing last-mile operations, and retains the lowest-cost verified solution among its recorded incumbents. Each learning-based policy samples 30 candidates per instance. **Reporting.** Metrics and timing follow Section 5. Quality comparisons include only configurations feasible on all 500 instances; other quality entries are dashed, with FR reported separately. Reported mean online time includes failures. Cost, distance, and time use USD, km, and seconds; bold marks the lowest mean cost within each stated comparison.

6.1 Deploying Euclidean-Trained Policies on Road Networks

Small-scale Euclidean training can yield feasible and competitive road-network solutions, but the outcome depends on both the policy and the deployment scale (Table 2). Four Euclid-100 policies remain fully feasible across all three scales, whereas TERRAN achieves FR of 93.0%, 8.6%, and 14.8% at Cus100, Cus500, and Cus1000; its quality entries are therefore omitted under our reporting rule. VNS/TS consolidates only part of its routes within the budget at Cus500 and Cus1000, yielding large fleets (Table 18 reports its cost from 60 to 1,800 seconds). All four fully feasible policies trail ALNS in mean cost at Cus100, but RRNCO and DRL-TS reverse this ordering at larger workloads: at Cus1000, their mean costs are 30.1% and 28.1% lower than ALNS's, respectively. The large-scale advantage reflects a trade-off between fleet size and travel distance. At Cus1000, RRNCO and DRL-TS use about 11 vehicles against ALNS's 17, but drive 2.41 and 2.65 times its distance. Under the reference calibration, the vehicle fixed-charge savings outweigh the additional distance cost. We next examine whether road-network training makes feasible, competitive deployment more consistent across policies (Section 6.2).

6.2 Training on Road Instances and Matching the Deployment Scale

Road training extends competitive large-scale deployment from two policies to all five (Table 3). All Road-100 policies cost less than ALNS at Cus500 and Cus1000 and remain feasible at all four deployment scales. At Cus1000, switching from Euclid-100 to Road-100 reduces mean cost by 31.4–65.6% for the four architectures with fully feasible Euclid-100 baselines. These gains support

Table 2: **Euclid-100 deployment versus search on T1.** †: FR at Cus100/500/1000 is 100/99.2/0.4% for Gurobi (1,800 s, two station copies) and 93.0/8.6/14.8% for TERRAN; otherwise 100%. Quality entries are dashed below 100% FR. Search uses the 1,800-second budget.

	Cus100				Cus500				Cus1000			
Method	Cost	K	D	Time	Cost	K	D	Time	Cost	K	D	Time
Gurobi†	515.94	1.18	178.08	—	—	—	—	—	—	—	—	—
ALNS	**451.21**	1.03	154.97	—	2,622.22	6.08	701.86	—	7,177.41	16.76	1,624.84	—
VNS/TS	518.23	1.19	165.94	—	88,533.71	209.57	12,181.85	—	288,943.07	683.34	41,456.02	—
AM-EVRPTW	887.28	1.92	613.55	0.55	4,640.61	9.62	4,369.76	4.34	11,253.09	23.15	11,054.20	10.33
EVRPTW-RL	689.56	1.50	455.41	0.65	3,872.22	7.98	3,765.56	12.91	8,996.59	18.33	9,317.05	46.43
DRL-TS	528.87	1.16	323.24	1.88	2,484.32	5.29	1,951.92	3.31	5,159.17	10.89	4,308.89	8.72
TERRAN†	—	—	—	0.16	—	—	—	1.66	—	—	—	10.07
RRNCO	803.72	1.81	373.61	0.10	**2,300.21**	4.94	1,681.76	0.85	**5,020.36**	10.70	3,922.96	5.59

training on the target environment, although they cannot be attributed to road geometry alone: the corpora also differ in delivery attributes and station layouts. The lowest-cost checkpoint depends on the deployment scale. Among the evaluated configurations, matching training and deployment scales gives the lowest mean cost for every architecture at Cus100, Cus500, and Cus1000; the final training stage therefore does not uniformly replace the earlier checkpoints. At the unseen Cus2000 scale, Road-1000 performs best for four architectures, while AM-EVRPTW performs best with Road-500. These are comparisons of complete training configurations: later stages accumulate more training and Cus100 uses a smaller station roster (Section 4.2), so the pattern does not isolate customer count alone. Because these comparisons use the training cities and communities, we next test whether the Road-1000 checkpoints remain feasible and competitive in geographically held-out regions (Section 6.3).

Table 3: **Road training across deployment scales.** Bold: lowest mean cost.

		Deployment scale: mean cost (USD)			
Method	Training	Cus100	Cus500	Cus1000	Cus2000
AM-EVRPTW	100	**468.00**	1,864.82	3,867.03	8,935.55
	500	504.84	**1,793.22**	3,831.96	**7,955.57**
	1,000	561.74	1,926.84	**3,781.91**	8,080.43
EVRPTW-RL	100	**481.76**	1,778.03	3,517.81	7,589.65
	500	508.84	**1,632.86**	3,189.34	6,581.58
	1,000	552.47	1,680.92	**3,171.85**	**6,554.95**
DRL-TS	100	**445.80**	1,225.98	2,073.61	3,776.48
	500	446.71	**1,197.15**	2,012.15	3,670.99
	1,000	463.86	1,220.76	**1,997.51**	**3,656.73**
TERRAN	100	**462.10**	1,720.15	3,722.23	10,271.72
	500	513.55	**1,343.06**	3,156.64	6,263.25
	1,000	465.43	1,423.40	**2,647.91**	**4,872.44**
RRNCO	100	**454.48**	1,605.39	3,441.93	8,027.42
	500	471.30	**1,321.30**	2,472.53	5,292.81
	1,000	569.70	1,344.74	**2,325.52**	**4,668.13**

6.3 Transfer to Held-Out Communities and an Unseen City

We deploy the same Road-1000 checkpoints at Cus1000 without retraining, in held-out communities (T2) and an unseen city (T3). All five policies remain fully feasible and achieve lower mean cost than ALNS on both held-out cohorts (Table 4). For every policy, mean cost as a percentage of ALNS's is lower on T2 than on T1. On T3, this ratio stays within 1.4 percentage points of its T1 value, despite higher absolute costs for both the learned policies and ALNS. These results support geographic transfer on the evaluated held-out cohorts. They do not identify which road-network features guide the policies' decisions; differences between instance populations also prevent isolating geographic transfer from instance difficulty. Additional solver results, fleet sizes, and distances appear in Appendix F.5. We next return to Cus1000 T1 to assess how much additional instance-specific search can improve the learned solutions (Section 6.4).

Table 4: **Road-1000 policies at Cus1000 across geographic splits.** Mean cost (USD) over 500 instances per split; all displayed configurations have 100% FR. T1: new delivery days; T2: held-out communities; T3: unseen city. % ALNS: mean cost as a percentage of ALNS's on the same split.

	T1		T2		T3	
Method	Cost	% ALNS	Cost	% ALNS	Cost	% ALNS
ALNS	7,177.41	100.0	7,815.28	100.0	8,756.74	100.0
AM-EVRPTW	3,781.91	52.7	3,599.54	46.1	4,591.52	52.4
EVRPTW-RL	3,171.85	44.2	2,996.83	38.3	3,988.55	45.5
DRL-TS	**1,997.51**	27.8	**1,969.09**	25.2	**2,344.66**	26.8
TERRAN	2,647.91	36.9	2,536.03	32.4	3,299.72	37.7
RRNCO	2,325.52	32.4	2,262.48	28.9	2,821.35	32.2

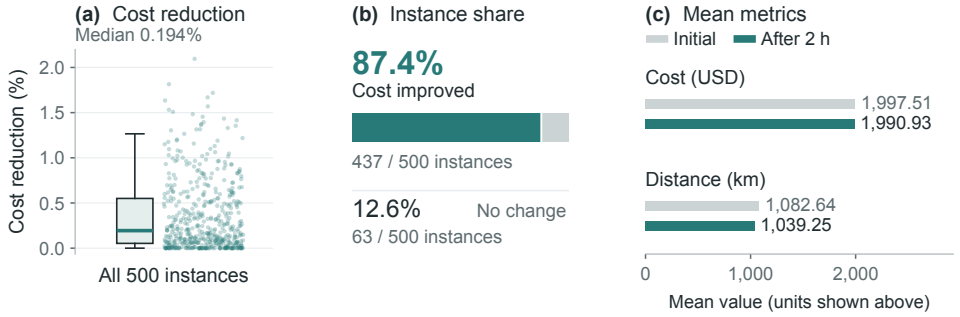


Figure 2: **Two-hour ALNS refinement on 500 Cus1000 T1 instances.** (a) Per-instance cost reductions, shown as points and a boxplot. (b) Shares of instances with reduced or unchanged cost. (c) Mean cost and distance before and after refinement.

6.4 How Much Cost Remains Recoverable

For each of the 500 Cus1000 T1 instances, we select the lowest-cost verified solution returned by the five fixed Road-1000 policies. DRL-TS provides this selected solution on every instance. We then use each selected plan to warm-start a separate ALNS run with two additional hours of search. This post-inference refinement uses the same-scale learned reference, and its outputs are not added to the benchmark reference pool. ALNS refinement improves most instances, but the recovered cost is small. For each instance, we measure the percentage cost reduction relative to its own starting plan. Across all 500 instances, including unchanged plans, the median reduction is 0.194% (interquartile range 0.053–0.551%; Figure 2(a)). Costs decrease on 437 instances (87.4%) and remain unchanged on 63 (12.6%; Figure 2(b)). Mean cost falls from \$1,997.51 to \$1,990.93, a reduction of \$6.58 per instance, or 0.329% of the initial mean (Figure 2(c)). All gains come from the distance term: fleet size remains unchanged on every instance, averaging 4.432 vehicles, while mean distance falls from 1,082.64 to 1,039.25 km. The tested two-hour ALNS refinement therefore recovers little additional cost from these DRL-TS solutions; this establishes neither cost optimality nor minimal fleet size (Appendix F.6).

7 Conclusion

We introduced EVRP-TW-D/B, combining a road-grounded corpus and reusable generator with a shared benchmark across solver families. The suite enables reproducible evaluation across operational workloads and geographies. Our experiments show that competitiveness after small-scale Euclidean training varies across policies and workload scales. Under the reference calibration, training the same architectures on 100-customer EVRP-TW-D instances yields full feasibility and lower mean costs than the strongest evaluated search baseline under a 30-minute budget at 500 and 1,000 customers. Policies trained at 1,000 customers retain this advantage beyond the training geography. These findings highlight the need to assess algorithmic progress across training and deployment conditions. The released suite provides a foundation for developing stronger routing algorithms and extending operational models while preserving verifiable, comparable results.

AI USE STATEMENT

We used OpenAI Codex to identify related datasets and literature, assist with experimental design and result interpretation, and draft and edit manuscript text. The authors consulted the original publications and reviewed the resulting text and analyses. Codex also assisted with reimplementing baselines from published descriptions, adapting available CVRP implementations to EVRP-TW, integrating dataset inputs into training pipelines, and improving instance-generation efficiency through multiprocessing. Dataset instances were produced by the procedural generator; none were directly generated by an LLM.

We checked baseline architectures and hyperparameters against the details reported in the source papers and reviewed and verified the AI-assisted code. Where papers omitted implementation details, Codex assisted in completing those details, subject to author review. These choices do not establish exact reproduction of the original implementations. The authors take responsibility for the final manuscript, implementations, dataset, and reported results.

REPRODUCIBILITY STATEMENT

An anonymized code repository accompanies this submission at `https://anonymous.4open.science/r/EVRPTW_DB_ICLR-67BD/`, providing the two-stage generator, independent verifier, and implementations of the evaluated solvers. Because the complete dataset exceeds 100 GB, the supplementary ZIP includes runnable code, pretrained checkpoints, and a small sample dataset for testing, with environment setup instructions and demonstrations of training, evaluation, and route visualization. The complete dataset will be released publicly on Hugging Face with the camera-ready version. The task and its formulations are specified in Appendix A, the cost calibration in Appendix B, dataset construction in Appendix C, and training configurations, budgets, hardware, and reporting rules in Appendix E.

ETHICS STATEMENT

Delivery attributes are derived from the Amazon Last Mile Routing Research Challenge data (Merchán et al., 2024), whose documentation states that personally identifiable information is excluded. No Amazon coordinates are transferred, and assigned delivery attributes do not describe actual deliveries to the selected residential locations or their occupants (Appendix D). Other inputs come from public geographic and infrastructure data. Source provenance and component-specific licensing terms are documented in Appendix D.3.

REFERENCES

Mehmet Anil Akbay and Christian Blum. EVRPGen: A web-based instance generator for the electric vehicle routing problem with road junctions and road types. *Software Impacts*, 25:100778, 2025. doi: 10.1016/j.simpa.2025.100778. URL https://doi.org/10.1016/j.simpa.2025.100778.

Hussein Basma, Felipe Rodríguez, Julia Hildermeier, and Andreas Jahn. Electrifying last-mile delivery: A total cost of ownership comparison of battery-electric and diesel trucks in europe. 2022.

Xiaowen Bi, Ruoheng Wang, and Qiang Jia. On the speed-varying range of electric vehicles in time-windowed routing problems with en-route partial re-charging. *IEEE Transactions on Consumer Electronics*, 70(1):3650–3657, 2023.

Jannis Blauth, Stephan Held, Dirk Müller, Niklas Schlomberg, Vera Traub, Thorben Tröbst, and Jens Vygen. Vehicle routing with time-dependent travel times: Theory, practice, and benchmarks. *Discrete Optimization*, 53:100848, 2024.

Maurizio Bruglieri, Bülent Çatay, Merve Keskin, Simona Mancini, and Ornella Pisacane. The mixed-fleet vehicle routing problem with low emission zones. *Transportation Research Part E: Logistics and Transportation Review*, 201:104230, 2025.

Andrew Burnham, David Gohlke, Luke Rush, Thomas Stephens, Yan Zhou, Mark A. Delucchi, Alicia Birky, Chad Hunter, Zhenhong Lin, Shiqi Ou, Fei Xie, Camron Proctor, Steven Wiryadinata, Nawei Liu, and Madhur Boloor. Comprehensive total cost of ownership quantification for vehicles with different size classes and powertrains. Technical Report ANL/ESD-21/4, Argonne National Laboratory, 2021. URL https://publications.anl.gov/anlpubs/2021/05/167399.pdf. Corrected version incorporating July 2021 erratum.

Cristian Cataldo-Díaz, Rodrigo Linfati, and John Willmer Escobar. Mathematical models for the electric vehicle routing problem with time windows considering different aspects of the charging process. *Operational Research*, 24(1):1, 2024.

Jinbiao Chen, Huanhuan Huang, Zizhen Zhang, and Jiahai Wang. Deep reinforcement learning with two-stage training strategy for practical electric vehicle routing problem with time windows. In *International conference on parallel problem solving from nature*, pp. 356–370. Springer, 2022.

Mertcan Daysalilar, Fuat Uyguroglu, Gabriel Nicolosi, and Adam Meyers. SynthCharge: An electric vehicle routing instance generator with feasibility screening to enable learning-based optimization and benchmarking. *arXiv preprint arXiv:2603.03230*, 2026. doi: 10.48550/arXiv.2603.03230. URL https://arxiv.org/abs/2603.03230.

Guy Desaulniers, Fausto Errico, Stefan Irnich, and Michael Schneider. Exact algorithms for electric vehicle-routing problems with time windows. *Operations Research*, 64(6):1388–1405, 2016.

Ya-ru Duan, Yong-shi Hu, and Peng Wu. An adaptive large neighborhood search heuristic for the electric vehicle routing problems with time windows and recharging strategies. *Journal of Advanced Transportation*, 2023(1):1200526, 2023.

Federal Highway Administration. Highway Performance Monitoring System (HPMS). https://www.fhwa.dot.gov/policyinformation/hpms.cfm, 2026. Program overview, updated 3 September 2026. Accessed 26 September 2026.

Dominik Goeke. Granular tabu search for the pickup and delivery problem with time windows and electric vehicles. *European Journal of Operational Research*, 278(3):821–836, 2019.

Dominik Goeke and Michael Schneider. Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research*, 245(1):81–99, 2015.

Ahmed Heakl, Yahia Salaheldin Shaaban, Salem Lahlou, Martin Takac, and Zangir Iklassov. SVRPBench: A realistic benchmark for stochastic vehicle routing problem. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3983. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/ad0e5a0e026f1b0ff259140564d18680-Abstract-Datasets_and_Benchmarks_Track.html. Datasets and Benchmarks Track.

Gerhard Hiermann, Jakob Puchinger, Stefan Ropke, and Richard F Hartl. The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *European Journal of Operational Research*, 252(3):995–1018, 2016.

Chad Hunter, Michael Penev, Evan Reznicek, Jason Lustbader, Alicia Birky, and Chen Zhang. Spatial and temporal analysis of the total cost of ownership for class 8 tractors and class 4 parcel delivery trucks. Technical Report NREL/TP-5400-71796, National Renewable Energy Laboratory, 2021. URL https://docs.nlr.gov/docs/fy21osti/71796.pdf.

IONNA. Bell, california rechargery relay—atlantic avenue. https://www.ionna.com/rechargery/bell-ca-rechargery-relay-atlantic-ave/. Published standard price of 0.39 USD/kWh, plus tax; temporary promotions excluded. Accessed 7 September 2026.

Merve Keskin and Bülent Çatay. Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation research part C: emerging technologies*, 65:111–127, 2016.

Merve Keskin Özel. *Recharge strategies for the electric vehicle routing problem with time windows in deterministic and stochastic environments*. PhD thesis, 2018.

Wouter Kool, Herke Van Hoof, and Max Welling. Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475*, 2018.

Yeong-Dae Kwon, Jinho Choo, Byoungjip Kim, Iljoo Yoon, Youngjune Gwon, and Seungjai Min. Pomo: Policy optimization with multiple optima for reinforcement learning. *Advances in neural information processing systems*, 33:21188–21198, 2020.

Edward Lam, Guy Desaulniers, and Peter J Stuckey. Branch-and-cut-and-price for the electric vehicle routing problem with time windows, piecewise-linear recharging and capacitated recharging stations. *Computers & Operations Research*, 145:105870, 2022.

Gilbert Laporte. What you should know about the vehicle routing problem. *Naval Research Logistics (NRL)*, 54(8):811–819, 2007.

Lelit Logistics LLC. Delivery driver opportunities. https://lelitlogistics.com/. Employer-advertised ten-hour daily schedule. Accessed 7 September 2026.

Bo Lin, Bissan Ghaddar, and Jatin Nathwani. Deep reinforcement learning for the electric vehicle routing problem with time windows. *IEEE transactions on intelligent transportation systems*, 23 (8):11528–11538, 2021.

Daniel Merchán, Jatin Arora, Julian Pachon, Karthik Konduri, Matthias Winkenbach, Steven Parks, and Joseph Noszek. 2021 Amazon last mile routing research challenge: Data set. *Transportation Science*, 58(1):8–11, 2024. doi: 10.1287/trsc.2022.1173. URL https://doi.org/10.1287/trsc.2022.1173.

Microsoft Maps. United States Building Footprints. https://github.com/microsoft/USBuildingFootprints, 2018. First released in 2018; repository accessed 26 September 2026.

Nima Moradi, Vahid Kayvanfar, and Roberto Baldacci. Electric-vehicle routing problem with time windows and energy minimization: green logistics with same-day delivery approaches. In *2024 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pp. 1–6. IEEE, 2024.

Arash Mozhdehi, Mahdi Mohammadizadeh, and Xin Wang. Edge-direct: A deep reinforcement learning-based method for solving heterogeneous electric vehicle routing problem with time window constraints. In *Canadian AI*, 2024.

Gaute Messel Nafstad, Guy Desaulniers, and Magnus Stålhane. Branch-price-and-cut for the electric vehicle routing problem with heterogeneous recharging technologies and nonlinear recharging functions. *Transportation Science*, 59(3):628–646, 2025.

National Laboratory of the Rockies. Alternative Fuels Data Center: Data Downloads. https://afdc.energy.gov/data_download, n.d. U.S. Department of Energy alternative fuel station data, continuously updated. Accessed 26 September 2026.

National Renewable Energy Laboratory. 2024 transportation annual technology baseline: Levelized cost of driving assumptions. https://atb.nlr.gov/transportation/2024/levelized_cost_of_driving_assumptions, 2024. Accessed 7 September 2026.

OpenStreetMap contributors. OpenStreetMap. https://www.openstreetmap.org/copyright, n.d. Open geospatial database. Accessed 26 September 2026.

Pacific Gas and Electric Company. Ev fleet: Total cost of ownership—distribution and delivery. https://www.pge.com/content/dam/pge/docs/clean-energy/electric-vehicles/PGE-EV-Fleet_Total-Cost-of-Ownership-Distribution-Delivery.pdf.coredownload.pdf, 2022. Reference duty-cycle inputs on p. 2; not a Rivian-specific estimate.

Sergio Maria Patella, Gianluca Grazieschi, Valerio Gatta, Edoardo Marcucci, and Stefano Carrese. The adoption of green vehicles in last mile logistics: A systematic review. *Sustainability*, 13(1): 6, 2020.

Dongbo Peng, Kanok Boriboonsomsin, and Matthew Barth. Routing short-haul trucks under the uncertainties of travel time and service time. Dryad, 2025. URL https://doi.org/10.5061/dryad.6hdr7srbv. Dataset.

Progressive Commercial. Commercial auto insurance. https://www.progressivecommercial.com/commercial-auto-insurance/. 2025 new-policy mean premium per power unit, business auto, clean driving records; liability and physical damage. Accessed 7 September 2026.

Sina Rastani, Merve Keskin, Tuğçe Yüksel, and Bülent Çatay. Uphill struggles, downhill gains: How road gradients and load dynamics influence electric vehicle routing decisions? *European Journal of Operational Research*, 2026.

Rivian. Rivian commercial van: Fleet. https://rivian.com/fleet. Delivery 700 starting price, before additional taxes and fees. Accessed 7 September 2026.

Rivian. Rivian commercial van 500 — 700: Reference guide, 2025 model year. https://assets.ctfassets.net/2md5qhoeajym/5FQcJgfAOa4vDYu9rWwEYO/4de1a8dc14ab58f3f21009c1d3df5fd2/RCV-QuickRef-v17.pdf, 2025. Version 17; weight values on p. 4 are marked pre-production.

Michael Schneider, Andreas Stenger, and Dominik Goeke. The electric vehicle routing problem with time windows and recharging stations. Technical Report 02/2012, University of Kaiserslautern, Chair of Business Information Systems and Operations Research, 2012. URL https://www.eflab.de/publications/2012.

Michael Schneider, Andreas Stenger, and Dominik Goeke. The electric vehicle-routing problem with time windows and recharging stations. *Transportation science*, 48(4):500–520, 2014.

Marius M Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265, 1987.

Jiwoo Son, Zhikai Zhao, Federico Berto, Chuanbo Hua, Zhiguang Cao, Changhyun Kwon, and Jinkyoo Park. RRNCO: Towards real-world routing with neural combinatorial optimization. In *International Conference on Learning Representations*, volume 2026, pp. 91794–91818, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/9446c291a8744a125a0bda5b18f4d5a1-Abstract-Conference.html.

Maojie Tang, Nanpeng Yu, Ioannis Karamouzas, and Zuzhao Ye. Terran: A transformer-based electric vehicle routing agent for real-time adaptive navigation. *IEEE Transactions on Automation Science and Engineering*, 2025.

Paolo Toth and Daniele Vigo. *The vehicle routing problem*. SIAM, 2002.

U.S. Army Corps of Engineers. National Structure Inventory 2026 Base: Technical Documentation. https://www.hec.usace.army.mil/confluence/nsi/technicalreferences/2026/technical-documentation, 2026. NSI 2026 Base documentation. Accessed 26 September 2026.

U.S. Bureau of Labor Statistics. Summary of annual and semi-annual consumer price indexes. https://www.bls.gov/regions/mid-atlantic/data/ConsumerPriceIndexAnnualandSemiAnnual_Table.htm. CPI-U U.S. city average: 2019, 255.657; 2022, 292.655; 2025, 321.943. Accessed 7 September 2026.

U.S. Bureau of Labor Statistics. Employer costs for employee compensation—june 2025. https://www.bls.gov/news.release/archives/ecec_09122025.htm, 2025a. Released 12 September 2025; Table 4, private transportation and warehousing.

U.S. Bureau of Labor Statistics. Occupational employment and wages—may 2024. https://www.bls.gov/news.release/archives/ocwage_04022025.htm, 2025b. Released 2 April 2025; Table 1, light truck drivers, SOC 53-3033.

U.S. Census Bureau. 2025 TIGER/Line Shapefiles. https://www2.census.gov/geo/tiger/TIGER2025/, 2025. Released 23 September 2025. Accessed 26 September 2026.

U.S. Environmental Protection Agency. Motor Vehicle Emission Simulator: MOVES5. https://www.epa.gov/moves/latest-version-motor-vehicle-emission-simulator-moves, November 2024. Office of Transportation and Air Quality, Ann Arbor, MI.

Jieman Xia, Zhou He, Shuolei Wang, Siliang Liu, and Shuai Zhang. Mean–standard-deviation-based electric vehicle routing problem with time windows using Lagrangian relaxation and extended alternating direction method of multipliers-based decomposition algorithm. *Soft Computing*, 28 (11–12):7139–7160, 2024. doi: 10.1007/s00500-023-09599-3.

A Problem Formulations and Decision Interfaces

The task in Section 3 admits a mixed-integer formulation that selects routes jointly and a sequential formulation that constructs them one visit at a time. Both use the same resource dynamics and minimum-cost objective. The formulations below distinguish these task requirements from implementation restrictions, including finite station copies, action masks, and decoding limits. Calibration and method-specific configurations are given in Appendices B and E.

A.1 Common Quantities and Route Semantics

Instance and resources. Let $\mathcal{C} = \{1, \dots, N\}$ be the customers, $\mathcal{S}$ the physical charging stations, and o the depot. A customer i has volume q_i, service duration h_i, and a hard service-start window $[a_i, \ell_i]$. Homogeneous vehicles have cargo capacity Q and usable battery capacity B. Each vehicle starts at the depot opening time t_0 with a full battery and returns by t_1. All stations in the instance are available over this horizon, without queuing. Time is measured in seconds, distance in kilometers, energy in kWh, and power in kW. The released records store an absolute clock within the day; subtracting t_0 from both time variables and customer-window endpoints gives the zero-origin horizon $[0, T_{\max}]$ of Section 3, with $T_{\max} = t_1 - t_0$ (Appendix C.5). We keep t_0 and t_1 below to match the released records.

Travel quantities. For every ordered pair of terminals, define

$$d_{ij} = d^{\mathrm{rt}}_{ij}, \qquad t_{ij} = t^{\mathrm{rt}}_{ij}, \qquad e_{ij} = \kappa d_{ij}. \tag{1}$$

Here t^{rt}_{ij} is the travel time of a fastest-time road path and d^{rt}_{ij} is the length of that same path, as in Section 3. The reference task uses this single path criterion: scheduling follows t, while both the cost distance and battery propagation use d. These directed quantities are static within an instance.

Shortest-distance quantities are retained in the release as auxiliary records and enter neither the objective nor any constraint (Appendix C.5).

A station s has stored power p_s^{cap}, capped by the vehicle's charging limit. Its battery-side effective power is $P_s^{\mathrm{eff}} = \eta p_s^{\mathrm{cap}}$, where the charging-power derating factor η is applied once. This is the effective power used in Section 3. A visit arriving with battery b takes

$$g_s(b) = \frac{3600(B-b)}{P_s^{\mathrm{eff}}} \tag{2}$$

seconds and leaves with battery B. No charging occurs at customers. Station power controls elapsed charging time; it does not change the reference price coefficient.

Feasible plans and reference cost. A plan x is a collection of nonempty routes $r = (o, v_1, \ldots, v_L, o)$, with no internal depot visit. Every customer occurs exactly once across the collection, while physical stations may occur repeatedly. A vehicle executes one route, so $K(x)$ equals the number of deployed routes. Returning to the depot ends that vehicle's operation; another route represents another vehicle. There is no hard fleet-size limit. Any retained route serving only stations still incurs deployment and travel costs. A positive fixed deployment charge makes such a route dominated: deleting it reduces cost without changing customer service. Hence an optimal plan can be chosen with every route serving a customer and $K(x) \le N$; this is a dominance property, not an imposed fleet limit.

The total customer volume on each route is at most Q. Travel consumes e_{ij} before any charging at its destination. Battery energy remains nonnegative, service starts within each customer's window, and all travel, waiting, service, and charging fit before the route's depot return. Service may finish after a customer's window closes, provided its start is within that window and the route remains feasible.

Let $\mathcal{A}(x)$ be the multiset of route transitions, including station visits and depot returns. With $F = F_{\mathrm{day}}$ and $\alpha = p_e\kappa$, the common objective is

$$C(x) = FK(x) + \alpha D(x), \qquad D(x) = \sum_{(i,j)\in\mathcal{A}(x)} d_{ij}. \tag{3}$$

Repeated transitions are counted with multiplicity, so the modeled driving energy is exactly $\kappa D(x)$ and the travel term prices it at p_e. Appendix B specifies the numerical calibration.

A.2 Mixed-Integer Formulation

Expanded graph. The implemented arc-flow model uses separate start and end depot nodes, o^+ and o^-, and R_s copies of each physical station s. The implementation uses a common count $R_s = 2$ for all stations; its scope is discussed at the end of this subsection. Let $\widetilde{\mathcal{S}}$ contain these copies and let $\phi(v)$ map each model node to its physical terminal. Customers map to themselves, both depot nodes map to o, and station copies inherit their physical station's matrices and power. In this subsection, d_{ij}, t_{ij}, e_{ij} abbreviate the quantities indexed by $\phi(i), \phi(j)$. Put

$$\mathcal{V} = \{o^+, o^-\} \cup \mathcal{C} \cup \widetilde{\mathcal{S}}, \quad \mathcal{V}^+ = \mathcal{V} \setminus \{o^-\}, \quad \mathcal{V}^- = \mathcal{V} \setminus \{o^+\}. \tag{4}$$

The arc set $\mathcal{E} \subseteq \mathcal{V}^+ \times \mathcal{V}^-$ excludes self-loops, the empty-route arc (o^+, o^-), direct arcs between copies of the same physical station, nonfinite travel quantities, and arcs requiring more than a full battery. Direct consecutive visits to the same station add no recharge opportunity under static full-recharge semantics. Visits to different stations, and repeated visits to a physical station separated by other visits, remain expressible subject to the copy counts.

Variables and domains. The binary variable x_{ij} selects arc $(i,j) \in \mathcal{E}$. Continuous variables τ_i, u_i, β_i represent a node's scheduling time, a cumulative-delivery-volume resource, and its arrival battery. At a customer, τ_i is service-start time, not necessarily physical arrival time. At a station it is a time before charging; waiting may be represented by slack in time propagation. Domains are

$$a_i \le \tau_i \le \ell_i, \qquad 0 \le u_i \le Q, \qquad 0 \le \beta_i \le B, \tag{5}$$

where $[a_i, \ell_i] = [t_0, t_1]$ for depot and station nodes. Set $q_i = h_i = 0$ outside the customer set. Each station copy f has charging duration $g_f \ge 0$; put $g_i = 0$ at other nodes. The visit indicator v_f and fleet count K below are expressions in x, not additional independent binary decisions.

Coverage, flow, and objective. Write $\delta^-(i)$ and $\delta^+(i)$ for incoming and outgoing arcs. The model requires

$$\sum_{a\in\delta^-(c)} x_a = \sum_{a\in\delta^+(c)} x_a = 1, \qquad c \in \mathcal{C}, \tag{6}$$

$$v_f := \sum_{a\in\delta^-(f)} x_a = \sum_{a\in\delta^+(f)} x_a \le 1, \qquad f \in \widetilde{\mathcal{S}}, \tag{7}$$

$$K := \sum_{a\in\delta^+(o^+)} x_a = \sum_{a\in\delta^-(o^-)} x_a \ge 1. \tag{8}$$

The objective is

$$\min\ FK + \alpha \sum_{(i,j)\in\mathcal{E}} d_{ij}x_{ij}. \tag{9}$$

Thus a depot-to-station departure also incurs a vehicle charge; fleet cost is not counted only on direct depot-to-customer arcs. The single-visit condition in Equation (7) applies to each copy, not to the physical charging station.

Full recharge and initial resources. For a station copy f, let $G_f = 3600B/P^{\text{eff}}_{\phi(f)}$. Charging is imposed by

$$0 \le g_f \le G_f v_f, \tag{10}$$

$$-G_f(1-v_f) \le g_f - \frac{3600(B-\beta_f)}{P^{\text{eff}}_{\phi(f)}} \le G_f(1-v_f). \tag{11}$$

A visited copy therefore charges exactly the missing battery energy; an unused copy has zero charging duration. At the start depot,

$$\tau_{o^+} = t_0, \qquad u_{o^+} = 0, \qquad \beta_{o^+} = B. \tag{12}$$

The zero initial value of u is a resource-accounting convention. All deliveries are loaded physically before departure; u accumulates the demand assigned to the route rather than representing goods remaining on the vehicle.

Time and cargo propagation. For each $(i,j) \in \mathcal{E}$,

$$\tau_j \ge \tau_i + h_i + g_i + t_{ij} - M_t(1-x_{ij}), \tag{13}$$

$$u_j \ge u_i + q_j - M_u(1-x_{ij}). \tag{14}$$

Together with the variable bounds, these constraints enforce service-start windows, charging time, the operating horizon, and route-volume capacity. Their inequalities allow waiting and resource slack. The shared end-depot variables are upper bounds on the individual routes' return times and accumulated volumes; they do not require all vehicles to arrive at the same time or carry equal loads.

Battery propagation and depot return. Define departure battery by

$$\widehat{\beta}_i = \begin{cases} B, & i \in \{o^+\} \cup \widetilde{\mathcal{S}}, \\ \beta_i, & i \in \mathcal{C}. \end{cases} \tag{15}$$

For arcs whose destination is not o^-, impose both sides of the conditional equality:

$$-M_e(1-x_{ij}) \le \beta_j - \widehat{\beta}_i + e_{ij} \le M_e(1-x_{ij}), \qquad j \ne o^-. \tag{16}$$

Each selected return arc instead requires

$$\widehat{\beta}_i \ge e_{io^-} - M_e(1-x_{io^-}), \qquad (i,o^-) \in \mathcal{E}. \tag{17}$$

For a station predecessor, the latter is already ensured by the full-battery arc filter. There is deliberately no equality defining one common arrival battery at o^-: different vehicles may finish with different remaining energy.

Relaxation constants and connectivity. The constants relax only unselected arcs. The implementation chooses M_t from the horizon length, maximum finite arc time, maximum service duration, and maximum full-recharge duration; M_u from cargo capacity and total demand; and M_e from battery capacity and maximum finite arc energy, each with a positive numerical margin. These bounds retain the conditional equalities rather than replacing them by one-sided battery depletion.

Flow conservation alone does not exclude disconnected cycles. Summing Equation (13) around a cycle rules out positive total travel, service, or charging time; positive total customer demand likewise contradicts Equation (14). No separate subtour-cut family is introduced. Co-located stations may still admit a station-only cycle with zero duration. Such a component serves no customer and can be deleted without increasing cost or changing any depot-to-depot route. Station-only components can therefore be discarded when recovering the route set. Generalized inputs permitting customer-containing cycles with zero total demand and duration would require additional connectivity constraints.

Scope of the finite expansion. The copy budget R_s bounds visits to station s across the entire route set, not separately for each vehicle. The two-copy expansion keeps the model compact, while the common task imposes no station-visit count limit. This finite expansion may therefore exclude routes permitted by the common task, and we do not claim that this copy budget suffices for every instance. Bounds and optimality certificates apply to the instantiated model. Failure to return a verified-feasible plan does not establish infeasibility of the common task; submitted incumbents are independently verified against that task. Copy counts and computational budgets are recorded in Appendix E.

A.3 Sequential Decision Formulation

State and route construction. The sequential formulation builds routes for different vehicles in one episode. Let $\mathcal{I}$ denote the complete instance, and let Π_k contain all completed routes and the current route prefix after k decoding transitions. A sufficient state is

$$s_k = (\mathcal{I}, \Pi_k, i_k, \tau_k, u_k, b_k, k), \tag{18}$$

where i_k is the current terminal, τ_k the current vehicle's clock after service or charging, u_k its accumulated assigned customer volume, and b_k its remaining battery. The history identifies served customers, station visits on the current route, whether that route contains a customer, and the number of vehicle departures. It therefore also supports history-dependent implementation masks. The step index makes a finite decoder limit explicit. These variables define the environment state; a policy's normalized observations and feature encoding need not expose all of them directly.

An action selects the next terminal j. Initially the prefix is (o) and $(i_0, \tau_0, u_0, b_0) = (o, t_0, 0, B)$. A return to o closes the active route. If customers remain, construction continues with a different vehicle initialized at the same opening time t_0. This is sequential enumeration of separate daily routes, not a vehicle returning, reloading, and departing again.

Deterministic resource transitions. For a proposed transition from i_k to j, first compute

$$\widehat{\tau} = \tau_k + t_{i_k j}, \qquad \widehat{b} = b_k - e_{i_k j}. \tag{19}$$

The arc quantities must be finite and $\widehat{b} \geq 0$. Every executed action appends j to the route history and sets $i_{k+1} = j$. Subsequent resource updates depend on the destination.

For an unserved customer j, put $\sigma_j = \max\{\widehat{\tau}, a_j\}$. The necessary local conditions and transition are

$$u_k + q_j \leq Q, \qquad \sigma_j \leq \ell_j, \qquad \sigma_j + h_j \leq t_1, \tag{20}$$

$$(\tau_{k+1}, u_{k+1}, b_{k+1}) = (\sigma_j + h_j, u_k + q_j, \widehat{b}). \tag{21}$$

The customer is marked served. Waiting changes time but not the reference distance or energy quantities.

For a station j, full charging gives

$$(\tau_{k+1}, u_{k+1}, b_{k+1}) = \left(\widehat{\tau} + \frac{3600(B - \widehat{b})}{P_j^{\text{eff}}}, u_k, B\right), \qquad \tau_{k+1} \leq t_1. \tag{22}$$

The charger does not reset cargo or the vehicle's clock. The task permits subsequent visits to the same physical station; whether a particular decoder exposes such actions is a separate implementation choice.

For a depot return, the incoming travel and battery consumption are accounted for and $\widehat{\tau} \le t_1$ is checked before any reset. A nonempty completed route is appended to the route set. Resource variables reset to

$$(i_{k+1}, \tau_{k+1}, u_{k+1}, b_{k+1}) = (o, t_0, 0, B). \tag{23}$$

If customers remain, these values initialize a different vehicle; otherwise the completed episode enters an absorbing successful state. The reset itself does not deploy a vehicle or incur a fixed charge. No customer is served twice, and a depot self-loop does not deploy a vehicle. Any completed charger-only trip retains its departure and travel charges.

Masks and finite-horizon execution. The conditions above describe resource-valid local transitions, not a certificate that every remaining customer can still be served. The implemented hard environment adds a return-feasibility check, using station paths compatible with the current route's station restrictions. It also prevents a physical station from being revisited within that route; station availability is reset for the next vehicle. Some method adapters further disallow depot-to-station or consecutive station actions. These rules restrict construction and are documented as method-specific decisions, not added to the shared task definition. They are not a proof that every feasible canonical route remains reachable by every policy.

A rollout succeeds only after all customers have been served and an actual feasible depot return has closed the final route. An invalid action, an empty admissible-action set, or exhaustion of the configured step limit before completion terminates the rollout without a successful candidate. Failure of this construction procedure does not establish that the instance itself is infeasible. The finite step limit is a decoding restriction; the policy's success flag alone does not replace route-set acceptance.

Incremental cost and reward. For a resource-valid transition, define the deployment indicator $z_k = \mathbf{1}\{i_k = o,\ j \neq o\}$. The unshaped reward is

$$r_k = -\left(\alpha d_{i_k j} + F z_k\right). \tag{24}$$

The fixed charge is incurred on departure, including a departure to a station; return travel still contributes its distance charge. Returning to the depot does not incur a second fixed charge. On a complete feasible trajectory, each vehicle has exactly one departure, and with undiscounted return,

$$\sum_{k=0}^{H_x-1} r_k = -\alpha \sum_{k=0}^{H_x-1} d_{i_k i_{k+1}} - F \sum_{k=0}^{H_x-1} z_k = -C(x). \tag{25}$$

Here H_x is the number of executed transitions. A fixed positive normalization $Z_{\mathcal{I}}$ changes the identity to $\sum_k r_k / Z_{\mathcal{I}} = -C(x)/Z_{\mathcal{I}}$ and preserves solution ordering within that instance. Constraint penalties, success bonuses, and temporary soft-constraint training stages are auxiliary learning mechanisms; their raw returns are not the reported cost. Those choices are recorded per method in Appendix E.

A.4 Connection to Route-Set Evaluation

For MILP output, station copies are mapped back to physical stations, start/end depot copies to o, and station-only components are discarded. Sequential output is split at actual depot returns into distinct routes. The resulting candidate is evaluated under the same customer coverage, route structure, resource, charging, and horizon requirements. Its K and D are recomputed from those physical routes using Equation (3). Neither MILP auxiliary variables nor a decoder's accumulated reward determine the reported metrics.

The generator also constructs single-customer feasibility certificates on each view's final terminal set and charging-station roster, using the same t_{ij} and e_{ij}. Under the unlimited-fleet and no-queuing assumptions, the certified routes jointly establish that the instance admits a feasible plan (Appendix C.6). This existence guarantee concerns the common task; a finite station-copy model additionally needs enough copies to represent the required station visits.

Shared task semantics and cost accounting do not imply that finite station-copy models, restricted action masks, and bounded rollouts explore identical route sets. Their restrictions remain part of the method configurations used to interpret the benchmark comparisons.

B Operating-Cost Calibration

Accounting unit. The reference objective of Equation (3) combines a fixed vehicle–driver daily allocation with a distance-dependent energy charge. Each deployment commits one vehicle and one driver to a depot-to-depot trip. The fixed coefficient accounts for the vehicle resource and the driver's paid day rather than billing the actual duration of each route, and

$$F_{\mathrm{day}} = F_{\mathrm{veh}} + F_{\mathrm{drv}}, \qquad \alpha = p_e \kappa, \tag{26}$$

with distances in kilometers and cost in U.S. dollars. Because $D(x)$ accumulates the fastest-time path lengths actually travelled, $\alpha D(x)$ prices exactly the modeled battery energy $\kappa D(x)$ at p_e; no separate station-purchase bill is added. The calibration below fixes the relative price of fleet deployment and distance for the benchmark; it is not an estimate of one carrier's realized expenditure.

Vehicle ownership allocation. Vehicle daily cost annualizes capital expenditure and adds recurring ownership allowances. Let P_0 be the vehicle price before tax, τ an acquisition-tax rate, f_0 a one-time documentation allowance, L the ownership period, and S the resale value at its end. With $P = P_0(1+\tau) + f_0$ and annual capital rate r, the capital-recovery factor and equivalent annual charge are

$$a(r, L) = \begin{cases} \dfrac{r(1+r)^L}{(1+r)^L - 1}, & r > 0, \\ 1/L, & r = 0, \end{cases} \tag{27}$$

$$A_{\mathrm{cap}} = a(r, L)\left(P - \frac{S}{(1+r)^L}\right). \tag{28}$$

This credits resale once and prices capital commitment; no separate depreciation, loan-principal, or interest charge is added. With annual insurance, maintenance-and-repair, and registration allowances, the vehicle daily allocation is

$$F_{\mathrm{veh}} = \frac{A_{\mathrm{cap}} + A_{\mathrm{ins}} + A_{\mathrm{maint}} + A_{\mathrm{reg}}}{N_{\mathrm{op}}}, \tag{29}$$

where $N_{\mathrm{op}} = 300$ reference operating days per year. This is an ownership allocation, not a rental quotation or the immediate cash expense of dispatching an already-owned vehicle.

The reference uses a ten-year holding period and 90 miles per operating day, the latter from a published delivery-duty scenario (Pacific Gas and Electric Company, 2022), giving $M_{\mathrm{ann}} = 90 N_{\mathrm{op}} = 27{,}000$ miles/year. Transferring the Class-4 step-van age and mileage factors of Burnham et al. (2021) gives the resale proxy

$$S = P_0 (0.9342)^L (0.9996)^{M_{\mathrm{ann}} L/1000}. \tag{30}$$

The resale base excludes acquisition taxes and documentation fees. The upkeep allowance is $A_{\mathrm{maint}} = M_{\mathrm{ann}} c_m$, where c_m is a transferred Class-4 battery-electric maintenance-and-repair rate covering the combined category, so no second tire charge is added. Reference mileage calibrates these fixed allowances; it is neither a route-distance constraint nor an additional route-dependent maintenance bill.

Table 5 records the inputs and their scope. Historical dollar-valued proxies are converted from 2019 to 2025 dollars using $I = 321.943/255.657$ from the published BLS CPI-U series (U.S. Bureau of Labor Statistics); values published in 2024 and 2025 are used as published. The sources cover different vehicles, years, and settings; combining them defines a transparent reference scenario rather than a synchronized survey of delivery-fleet costs.

The insurance proxy concerns liability and physical-damage coverage for new business-auto policies with clean driving records; it is not a for-hire-truck quotation. Maintenance and registration use commercial-vehicle proxies, while documentation uses a light-duty proxy. The capital rate serves constant-price annualization. Carrier overhead, tolls, charging-infrastructure investment, subsidies, and separate battery-replacement events are not itemized in this reference model.

Table 5: Vehicle-cost inputs. Published quantities retain their source scope; proxies and reference assumptions are stated explicitly.

Input	Adopted value	Basis and interpretation
Vehicle price P_0	\$83,900	Delivery 700 published starting price, before tax and fees (Rivian).
Empty weight W	7,242 lb	Manufacturer reference-guide value marked pre-production (Rivian, 2025).
Holding period L; usable days N_{op}	10 years; 300/year	Reference ownership and utilization assumptions.
Reference mileage	90 miles/day	Transferred delivery-duty scenario (Pacific Gas and Electric Company, 2022).
Capital rate r	7%	Reference rate adopted from Transportation ATB assumptions (National Renewable Energy Laboratory, 2024).
Acquisition tax τ	5%	Common modeling scenario, not each city's tax rate (Hunter et al., 2021).
Documentation f_0	$300I$ dollars	Transferred light-duty documentation allowance (Burnham et al., 2021).
Resale S	Equation (30)	Class-4 step-van retention factors (Burnham et al., 2021).
Insurance A_{ins}	12×276 dollars/year	2025 business-auto new-policy mean, used as a proxy; not a Rivian quote (Progressive Commercial).
Upkeep c_m	$0.1813I$ dollars/mile	Transferred levelized Class-4 BEV maintenance-and-repair rate (Burnham et al., 2021).
Registration A_{reg}	$0.14WI$ dollars/year	Single-unit-truck empty-weight allowance; no added light-duty registration charge (Burnham et al., 2021).

Driver daily allocation. Driver cost is a fixed allocation based on a reference wage w, employer benefit burden b, and paid-day equivalent H_{ref}:

$$F_{\mathrm{drv}} = w(1+b)H_{\mathrm{ref}}. \tag{31}$$

The BLS May 2024 national mean wage for light-truck drivers is \$23.05/h, rounded to $w =$ \$23/h (U.S. Bureau of Labor Statistics, 2025b). For private transportation and warehousing, the BLS June 2025 estimates report \$33.75/h in wages and \$17.29/h in benefits, a ratio of approximately 51.23%, which we round to $b = 0.50$ (U.S. Bureau of Labor Statistics, 2025a). The sector-wide benefit ratio is a proxy for the reference driver, not a delivery-driver-specific measurement; benefits and payroll expenses already covered by this burden are not charged again.

We use $H_{\mathrm{ref}} = 10$ hours as a reference daily allocation, with publicly advertised delivery schedules providing an example of this workday length (Lelit Logistics LLC). The resulting driver charge is $23 \times 1.5 \times 10 = \$345$. This calibration introduces no ten-hour route-duration limit: feasibility still uses the instance's operating horizon. It also does not model overtime or bill waiting, travel, service, and charging hours separately. Those activities affect feasibility, while the objective charges the same daily allocation per deployment.

Electricity price and consumption rate. The uniform reference price $p_e = \$0.39$/kWh is anchored to the published standard tariff at IONNA's Bell, California station, before local tax and excluding temporary promotions (IONNA). It is neither a national mean nor the observed tariff of every selected station, and it is frozen in the objective profile rather than queried during evaluation. The rate $\kappa = 100/257$ kWh/km is a specification-based approximation from the Delivery 700's listed 100-kWh pack and 257-km estimated range (Rivian, 2025). The guide does not identify the listed pack capacity as usable capacity; treating 100 kWh as the usable resource is a benchmark assumption (Appendix C.5). Effective charging power, including its one-time derating, changes charging duration and feasibility; it does not rescale α. Station-dependent tariffs, time-of-use prices, and purchase-and-residual-energy settlement require a different objective and lie outside this calibration.

Numerical result and precision. The inputs give $P = \$88{,}472.78$ and $S = \$38{,}128.30$. Table 6 reports the resulting allocations. Totals use unrounded values, yielding

$$F_{\mathrm{veh}} = 68.6331536717643, \qquad F_{\mathrm{drv}} = 345, \tag{32}$$

$$F_{\mathrm{day}} = 413.6331536717643, \qquad \alpha = 0.39 \times \frac{100}{257} \simeq 0.151750972762646. \tag{33}$$

The corresponding profile is `rivian_energy_vehicle_cost_v2`. It stores the fixed charge, electricity price, and consumption rate; implementations multiply the stored price and rate rather than using rounded display coefficients. Instance-level battery-consumption parameters do not silently overwrite the economic profile.

Table 6: Derived vehicle and driver daily allocations in U.S. dollars. Displayed components are rounded; totals use unrounded inputs.

Component	Annual allocation	Per operating day
Capital recovery	9,836.90	32.79
Vehicle insurance	3,312.00	11.04
Maintenance and repair	6,164.29	20.55
Registration	1,276.76	4.26
Vehicle subtotal	20,589.95	68.63
Driver allocation	—	345.00
Vehicle–driver total	—	413.63

The ratio $F_{\mathrm{day}}/\alpha \simeq 2{,}726$ km is the distance whose energy cost equals one additional deployment. It does not impose a lexicographic fleet-first objective. Reward shaping and normalization used internally by a learned method remain separate from this reported cost. Multiplying the complete objective by one positive constant preserves its ordering, while scaling the fleet and distance terms independently changes the comparison task.

C Dataset Construction and Release Organization

EVRP-TW-D combines public road and facility information with observed delivery requirements to produce a semi-synthetic EVRP-TW corpus. Its city environments provide candidate locations and directed travel networks; delivery-day generation selects customers and assigns observed order tuples within these environments. This appendix specifies the construction, the checks required for retaining an instance family, and the relationship between the main benchmark and additional views in the release. Candidate locations are supported by public source evidence; they are not field-verified operational facilities.

C.1 Public Sources and Their Roles

The inputs serve different roles in the construction (Table 7). Geographic sources establish where travel and service can occur, infrastructure sources supply facility candidates, and Amazon records provide delivery-day structure and order attributes. Amazon coordinates are not transferred into the generated cities. The source artifact contains 6,112 routes and 898,391 usable stop tuples from 521 station-days at 17 stations, covering July 19 to August 26, 2018. Across these station-days, the median number of usable drop-off stops is 1,388, computed from the order-template pool before the additional spatial-envelope filtering used for structure templates. This is the operational reference used in Section 1. These records come from the Amazon Last Mile Routing Research Challenge's training-source artifact; the generation and evaluation partitions below are internal partitions of that artifact.

C.2 City Boundaries and Directed Road Connectivity

Service boundary and routing extent. The service boundary is the 2025 Census Place polygon minus Census area-water polygons. Candidate service and facility locations must be covered by this land-only geometry, including its boundary. The routing network has a separate extent: roads outside the city may be needed to connect trips that leave and re-enter it. The original city-road graph is clipped using the administrative Place boundary, while progressively larger extraction envelopes retain actual connecting roads. Bridges and other routing edges are therefore not removed merely because they cross water or the service boundary.

The OSM driving graph is a directed multigraph. It preserves one-way restrictions, parallel edges, road lengths, edge geometry, and source node/way references. OSMnx simplification represents intermediate shape points in edge geometry rather than preserving every original shape point as a graph node. Imported outside-city nodes and incident edges are marked for transit; terminal eligi-

Table 7: Public inputs and their construction roles.

Source	Role in the released construction
Census TIGER/Line (U.S. Census Bureau, 2025)	Place and area-water polygons define the land-only service boundary; Block Groups define spatial units.
OpenStreetMap (OpenStreetMap contributors, n.d.)	Directed road topology, geometry, lengths, speed tags, and depot evidence.
HPMS (Federal Highway Administration, 2026)	Matched road-class evidence and supplementary speed limits.
EPA MOVES5 (U.S. Environmental Protection Agency, 2024)	National road-class and day-type reference speed factors.
NSI and Microsoft footprints (U.S. Army Corps of Engineers, 2026; Microsoft Maps, 2018)	Residential-use evidence, modeled residential units, and building geometry.
AFDC (National Laboratory of the Rockies, n.d.)	Public charging-site candidates, charging modes, and power-imputation inputs.
Amazon Last Mile records (Merchán et al., 2024)	Source route-by-travel-time structure and joint stop-level delivery tuples.

bility is determined separately. No physical road is fabricated to bridge disconnected components. Virtual terminal-access connectors, described below, are a distinct layer.

Coverage and residual components. The extraction buffers are 0, 1, 2, 5, 10, and 20 km. Coverage is measured against the original city graph, so imported external roads do not increase its numerator merely by increasing the routing graph's size. Let V_c and $\mathcal{E}_c^{\mathrm{phys}}$ be the original city nodes and physical roads, and let L_c be the selected routing component. The normal acceptance conditions are

$$\frac{|V(L_c) \cap V_c|}{|V_c|} \geq 0.99, \qquad \frac{\ell(\mathcal{E}_c^{\mathrm{phys}} \text{ covered by } L_c)}{\ell(\mathcal{E}_c^{\mathrm{phys}})} \geq 0.995. \tag{34}$$

Physical length removes duplicate representations of opposing directions while retaining geometrically distinct parallel roads. Normal selection uses the first buffer satisfying both conditions.

After the extraction ladder is exhausted, residual weak components with strictly fewer than 100 nodes may be excluded from the effective coverage denominator and recorded in the release metadata. Among admissible fallback candidates, selection maximizes raw node coverage, then raw physical-road coverage, and finally prefers a smaller buffer. Jacksonville is the only city using this fallback: the selected 1-km buffer has raw node coverage 0.9885173 and road coverage 0.9942059. Its 132 excluded small components contain 447 nodes and approximately 49.19 km of physical roads; effective coverage is approximately one. The other ten cities meet the normal raw-coverage conditions. The released records distinguish these two acceptance modes.

Directed access. Weak connectivity alone does not establish that a terminal can be served and a vehicle can return. Candidate access is checked relative to the largest strongly connected component, using the inbound and outbound directions of the terminal's road projections. The two access directions may use different directed references. Locations that fail these checks are excluded from the eligible pool with recorded reasons. The reference road model retains directional topology and prohibits immediate U-turns; additional turn and signal delay parameters are zero.

C.3 Reference Road-Speed Construction

Legal-speed proxy. Each directed edge receives a speed estimate using directional OSM `maxspeed`, generic OSM `maxspeed`, a usable HPMS speed limit, and within-city imputation, in that order. Multiple parsed OSM values use the conservative minimum. The resulting estimate is used as a free-flow proxy, not as a measurement of uncongested traffic.

OSM and HPMS geometries are matched in a city-specific metric projection. Candidates require lateral distance at most 75 m, axial orientation difference at most 30°, and overlap at least 0.20

Table 8: Frozen MOVES5 reference factors. The displayed factors are rounded.

Urban road type	Low-flow reference (mph)	Weekday ρ	Weekend ρ
Restricted access	70	0.758790960	0.841455883
Unrestricted access	45	0.545059126	0.564078025

using a 25-m buffer; route conflicts are rejected. High-confidence matches require an unambiguous candidate, distance at most 25 m, overlap at least 0.50, orientation difference at most 15°, and either exact route agreement or distance at most 5 m with overlap at least 0.80. HPMS speed limits are used only from high-confidence matches on unambiguously one-way OSM segments under a direction-assignment rule. This rule does not independently verify the direction of the HPMS observation. Missing speed estimates are imputed by length-weighted medians, successively using the same highway class, operating mode, parent highway class, and city-wide observations when a narrower group has no observations.

Day-type speed factors. MOVES5 supplies national default profiles for Light Commercial Truck (source type 32), using urban restricted-access and urban unrestricted-access road types. Road-class evidence maps edges to these two categories; high-confidence HPMS functional classes take precedence, with OSM highway classes used otherwise. For road type r, day type d, hour h, and speed bin k, let p_{rdhk} denote the within-hour vehicle-hours-traveled (VHT) fraction, s_k the representative speed, and a_{rdh} the hourly vehicle-miles-traveled (VMT) fraction. The hourly and service-window effective speeds are

$$\bar{v}_{rdh} = \sum_k p_{rdhk} s_k, \qquad v_{rd}^{\text{eff}} = \frac{\sum_{h \in H} a_{rdh}}{\sum_{h \in H} a_{rdh}/\bar{v}_{rdh}}, \tag{35}$$

where H is 08:00–24:00. Thus the hourly speeds are combined through VMT-weighted harmonic averaging.

For each road type, the low-flow reference v_r^{low} is the 85th percentile of the speed-bin distribution pooled over weekend 06:00–10:00. Each hour contributes VHT proportional to $a_{r,\text{weekend},h}/\bar{v}_{r,\text{weekend},h}$, distributed among bins according to $p_{r,\text{weekend},h,k}$. The selected reference is the first bin representative at which cumulative VHT weight reaches 0.85. The day-type factor is $\rho_{rd} = v_{rd}^{\text{eff}}/v_r^{\text{low}}$; both day types share the same road-type denominator (Table 8).

Instance speeds and access connectors. The CLE stores $v_{e,d}^{\text{ref}} = v_e^{\text{legal}} \rho_{r(e),d}$. The release uses unit residual multipliers and no additional edge noise. Its instance speed is

$$v_{e,d}^{\text{inst}} = \min\{v_e^{\text{legal}}, \max\{5, v_{e,d}^{\text{ref}}\}\} \quad \text{in km/h.} \tag{36}$$

The legal cap takes precedence over the nominal 5-km/h floor; lower legal limits therefore remain lower. Terminal-access connectors use the city/day length-weighted median instance speed of delivery-access operating-mode edges, falling back to all edges when that mode is absent. This operating-mode classification is distinct from the two MOVES road types. The two directions of an access connector have equal length and speed.

An instance selects one weekday or weekend profile and keeps it static throughout the day. These factors transfer a national reference relationship to city roads; they are not city-specific traffic observations. MOVES5 does not determine the energy-consumption coefficient. Speeds may affect modeled energy by changing the selected fastest-time path and its length, while the distance-based energy rate remains constant.

C.4 Service and Facility Layers

Residential service candidates. Residential NSI records provide use and residential-unit evidence; Microsoft footprints provide building geometry. The eligible pool requires unique point-in-footprint containment (G1); nearby, area-compatible matches without unique containment (G2) are excluded. Matched records are aggregated by Microsoft building identifier, so multiple NSI records can contribute to one service location. The eligible pool contains 3,531,561 service locations across the 11 cities.

Table 9: Physical parameters and their interpretation in the release.

Quantity	Value or assumption
Usable battery B	100 kWh
Nominal range; energy consumption rate κ	257 km; $100/257$ kWh/km
Cargo volume capacity Q	18,500,000 cm^3
Stored operating horizon	$[28{,}800, 86{,}400]$ s (08:00–24:00)
Departure and station charging	Full battery; constant-power full recharge
Fleet; station capacity	No hard fleet-size limit; no queuing
Site-power medians: AC L2 / DC	6.5 / 200 kW
Vehicle acceptance caps: AC / DC	11 / 100 kW
Power derating factor η	0.90, applied once
Stored capped power: AC / DC	6.5 / 100 kW
Battery-side power: AC / DC	5.85 / 90 kW
Additional turn / signal delay	0 / 0 s

Location types distinguish houses, manufactured homes, and apartment buildings with 2–4, 5–19, or at least 20 modeled units. Positive NSI unit counts take precedence; otherwise occupancy-category lower bounds provide the modeled count. This evidence influences activation weights, not the number of packages assigned on a delivery day.

Depots and charging sites. Depot candidates are grouped by physical facility. Tier A requires logistics or dispatch evidence associated with a facility; Tier B contains warehouse/logistics proxies whose parcel-dispatch function is not confirmed. Both enter the candidate pool, while generic industrial sites and retail mailing counters in Tier C are excluded. Selection operates on facility groups and prefers a Tier-A access point within a selected group when available. Of 7,500 parent families, 356 use Tier A and 7,144 use Tier B. The depot layer therefore consists of evidence-based candidates rather than verified parcel depots.

Charging candidates are public AFDC electric sites listed as available and supporting AC Level 2 or DC fast charging. Coordinates come from an exact-address OSM charging point where available and otherwise from the original AFDC coordinate. Eligibility additionally requires coordinate evidence and suitable road access: retaining a raw coordinate alone is insufficient. The 5,314 eligible sites comprise 54 exact-address OSM matches and 5,260 Census-address-corroborated AFDC points. Address corroboration supports the source location but does not verify its exact on-site geometry. Connector descriptions are retained as evidence; the reference task assumes vehicles can use all selected AC Level 2 and DC fast sites.

Road attachment. A customer footprint is linked to the nearest eligible physical road geometry by its shortest connecting segment; a facility uses its source or resolved point. A projection splits the referenced directed road edges while preserving their direction. Equal-length bidirectional connectors join each service/facility location to that projection. The procedure does not add physical roads between customer locations. Access distances above 200 m for customers or 250 m for facilities are quality flags, not automatic exclusion thresholds. Eligibility depends on source evidence, valid attachment, and directed access; the released records retain both the flags and rejection reasons.

C.5 Reference Parameters, Clocks, and Travel Quantities

The fixed vehicle and charging parameters in Table 9 define the resource constraints used in generation and evaluation. They are reference modeling values; the nominal range calibrates a constant rate $\kappa = 100/257$ kWh/km without a speed-dependent or auxiliary-load term. Monetary coefficients are specified separately in Appendix B.

Charging power. Each station receives the frozen national median site power of its charging mode $m(s)$, capped by the vehicle's acceptance limit for that mode:

$$p_s^{\text{cap}} = \min\{p_{m(s)}^{\text{site}}, p_{\text{veh},m(s)}^{\max}\}, \qquad P_s^{\text{eff}} = \eta p_s^{\text{cap}}. \tag{37}$$

The medians use public, available U.S. AFDC charging-unit records, weighted by J1772 connector-port counts for AC and CCS counts for DC. The resulting battery-side powers are 5.85 and 90 kW,

Table 10: Stored travel quantities and their use in the reference task.

Quantity	Meaning	Reference-task use
t_{ij}^{time}	Time of a fastest-time path (s)	Scheduling
d_{ij}^{time}	Length of that time-selected path (km)	Cost distance and battery consumption
d_{ij}^{dist}	Length of a shortest-distance path (km)	Auxiliary record
t_{ij}^{dist}	Time along that distance-selected path (s)	Auxiliary record

according to mode. The stored field `effective_charging_power_kw` contains p_s^{cap}, before derating. A vehicle arriving with energy b requires $3600(B-b)/P_s^{\mathrm{eff}}$ seconds to recharge (Equation (2)); the factor η must not be applied twice.

Time coordinates. Files and route verification use absolute seconds within the day. Subtracting 28,800 s from the clock and from both endpoints of every customer window, $t' = t - 28{,}800$, $a_i' = a_i - 28{,}800$, and $\ell_i' = \ell_i - 28{,}800$, gives the zero-origin horizon $[0, T_{\max}]$ of Section 3 with $T_{\max} = 57{,}600$ s. Travel, service, and charging durations are unchanged. Observation normalization by a learned policy does not change this storage or resource-propagation convention.

Travel quantities. For each ordered terminal pair, the parent stores the four quantities in Table 10, all as float32 arrays. Paths are computed on the directed network including terminal access and the reference turn rules. The reference task uses the fastest-time path: t_{ij}^{time} and d_{ij}^{time} are the quantities written t_{ij}^{rt} and d_{ij}^{rt} in Section 3. The shortest-distance quantities are auxiliary records and enter neither the objective nor any constraint.

The energy matrix is derived at loading time as $e_{ij} = \kappa d_{ij}^{\mathrm{time}}$, and the cost distance is $D(x) = \sum_{(i,j)\in\mathcal{A}(x)} d_{ij}^{\mathrm{time}}$, where $\mathcal{A}(x)$ is the multiset of route transitions, including repeated station visits and depot returns. The reference cost $F_{\mathrm{day}}K(x) + p_e\kappa D(x)$ therefore prices exactly the modeled driving energy $\kappa D(x)$ along the paths used for feasibility (Appendix A).

C.6 Delivery-Day Instance Materialization

Throughout this appendix, a *parent* is the base instance of Section 4.1; its instance family comprises the views derived from it by nested partitioning.

C.6.1 Source Templates and Partitions

An order template preserves one usable Amazon drop-off stop as a joint tuple: package count, summed package volume, summed planned service duration, and the intersection of its specified package windows, clipped to 08:00–24:00. An unspecified window contributes no additional restriction beyond that horizon. Invalid dimensions, malformed service or window fields, an empty window intersection, and volume exceeding vehicle capacity cause exclusion with an attrition reason. A zero-width window is not rejected solely for having equal endpoints; subsequent feasibility checks still apply. The source artifact contains 6,112 delivery-station stops, exactly one per source route; these are not delivery stops and are excluded from the template pool, as are 24 stops with unparsable package dimensions.

A structure template uses the source day's stops to describe the allocation of workload across source route groups and depot-to-stop travel-time bands. Structure and order sources have distinct roles, but they are selected from the applicable track/day-type pool and may be the same station-day. They coincide in 771 of the 7,500 released families. All families, including the 500 Cus2000 parents, use single-day sources for both roles.

Complete source station-days are assigned before instance generation. The target 70/15/15 allocation is based on usable order-template mass, not the number of days (Table 11). The descriptive metric holdout reserves the entire stations DCH2, DLA9, and DSE2. These station-days are excluded from instance generation and reserved as an independent reference for spatial-distribution diagnostics, including nearest-neighbor and within-route travel times and variation across source days. The remaining training and evaluation sources are separated by station-day. Evaluation's 57

Table 11: Source partitions by complete station-day. Allocation targets refer to usable tuple mass.

Source cohort	Station-days	Usable tuples	Tuple share (%)
Generation training	358	628,742	69.98534
Generation evaluation	57	134,759	15.00004
Descriptive metric holdout	106	134,890	15.01462
Total	521	898,391	100

days are further partitioned into 13 validation, 11 T1, 11 T2, 14 T3, and 8 scalability days, so those tracks do not reuse the same source templates.

C.6.2 Communities and Depot–Template Compatibility

Customer candidates are grouped by city, Census Block Group, and road-anchor strongly connected component. Directed road edges crossing these groups induce community adjacency. Road communities with no eligible customers remain as zero-capacity transit communities, so they can connect growing delivery regions. Whole communities receive the training/held-out assignment. A deterministic search over 256 starts targets 20% of eligible locations for holdout and balances counts by location type and modeled-unit band. It does not balance the sum of residential units or package demand. Actual held-out location shares range from approximately 19.996% to 20.079% across cities.

Each family's generation plan records its city, track, customer pool, parent scale, station count, day type, identifiers, and deterministically derived random-number namespaces. Depot facility groups and compatible structure sources are examined in a reproducible order. A source must belong to the prescribed pool and day type and provide enough retained stops for the requested parent size.

For a selected source, let T_{env} be the linearly interpolated 99th percentile of its usable depot-to-stop travel times. Only source stops at or below that value contribute to its structure. The target territory must have bidirectional road access, outbound travel-time proxy at most T_{env}, and satisfy

$$\kappa\big(\widehat{d}_{oi} + \widehat{d}_{io}\big) \le B, \tag{38}$$

where the distances are inexpensive depot-star proxies. Their distance calculations use separate shortest-distance searches and time-prioritized access choices; they are neither the final d^{time} values nor a substitute for exact terminal routing. This preliminary screen does not certify that a final fastest-time round trip fits one battery; the final per-view certificates below establish feasibility. A depot–source pair must additionally supply the required candidates in every band and admit the complete region/customer assignment. The first pair passing these checks is retained; selection is not uniform over all compatible pairs.

C.6.3 Workload Quotas, Region Growth, and Customer Assignment

The retained source times define ten quantile-based bands, with the first boundary set to zero and the last to T_{env}. Their boundaries are stored as unnormalized travel durations in seconds and applied directly to target-city outbound travel-time proxies. No cross-city quantile normalization is applied. Repeated source times can make band counts unequal despite their quantile-based construction.

Let C_{rb} count source stops in route group r and band b, and let $C = (C_{rb})_{r,b}$ denote the resulting count matrix. With $M = \sum_{r,b} C_{rb}$, the matrix $N_{\mathrm{target}} C/M$ gives fractional allocation targets for a parent of size $N_{\mathrm{target}} \le M$. Largest-remainder rounding first determines integer row and column totals. Controlled matrix rounding then produces nonnegative integer quotas n_{rb} with those margins and total N_{target}. It preserves the rounded margins rather than claiming equality to the original real-valued proportions.

Positive-quota groups are processed in decreasing workload order. A group's weighted-median quota band guides seed selection, with neighboring-band fallback where necessary. The first seed maximizes candidate support in the chosen band. Later seeds maximize their minimum separation from existing seeds, using

$$d_{\mathrm{sym}}(a, c) = \tfrac{1}{2}\big(t^{\mathrm{comm}}_{a\to c} + t^{\mathrm{comm}}_{c\to a}\big), \tag{39}$$

where the two terms are directed shortest travel times on the community graph. Each region expands through actual incoming or outgoing community adjacency, prioritizing unmet band capacity and then crossing time with deterministic tie-breaking. Regions may share communities; they are not disjoint geographic polygons.

A global minimum-cost flow resolves competition for candidate locations. Each route-group/band cell supplies its quota; cell-to-customer and customer-to-sink arcs have unit capacity, so a customer is selected at most once. Activation preferences use

$$k_{ri} = \frac{-\log U_{ri}}{\max\{1, \omega_i\}}, \qquad U_{ri} \in (0,1), \tag{40}$$

where ω_i is modeled residential-unit evidence and U_{ri} is a seeded pseudorandom key specific to the region/customer pair. Costs use rounded $10^9 k_{ri}$. Larger residential buildings receive greater activation preference, while every selected building remains one service stop. Further region expansion can resolve insufficient shared capacity. Source groups organize where demand is activated; they do not specify the solver's fleet size or vehicle routes.

C.6.4 Charging Rosters and Exact Terminal Matrices

Charging candidates first pass directed depot-access checks. An infrastructure graph on the depot and all remaining candidates permits an arc when $\kappa d_{ij}^{\text{time}} \le B$. Candidates must be reachable from the depot and able to return in this graph, allowing intermediate charging at other candidate stations.

For depot o, active customer i, and eligible station s, define

$$\delta_{is} = \max\left\{0, \min\left[t_{os}^{\text{time}} + t_{si}^{\text{time}} - t_{oi}^{\text{time}}, t_{is}^{\text{time}} + t_{so}^{\text{time}} - t_{io}^{\text{time}}\right]\right\}. \tag{41}$$

This is the smaller additional road time from inserting the station on either leg of the single-customer depot round trip. It excludes charging duration. For a nonempty roster S, put $\Delta_i(S) = \min_{s \in S} \delta_{is}$ and

$$J(S) = \text{mean}_i\, \Delta_i(S) + 0.25\ \text{quantile}_{0.90}\big(\{\Delta_i(S)\}_i\big) + 0.10\ \max_i \Delta_i(S). \tag{42}$$

Starting from the empty roster, each step chooses the remaining station minimizing $J(S \cup \{s\})$, updates every customer's minimum, and continues to the prescribed count. Seeds break equal-score ties. An insufficient candidate pool causes failure rather than a reduced station count. Candidate-level energy connectivity is not assumed to survive station removal: acceptance is checked again on each final view. The selected terminals define the exact four-matrix parent bundle in Table 10.

C.6.5 Order Matching and Single-Customer Certificates

For the selected parent, the certificate routine uses only its exact terminal matrices and retained charging stations. A depot or a station departed after full recharge is represented as a full-battery state. Feasible infrastructure transitions include travel and, at an arriving station, the time required to restore the energy consumed. Forward and reverse shortest-path calculations allow multiple charging stops before and after a customer. The incoming and outgoing customer legs must jointly fit between successive full-battery states, because service does not reset battery energy. If the next full-battery state is a station, its charging duration restores the energy consumed on both customer-adjacent legs.

The routine first chooses a single-customer certificate minimizing travel-and-charging duration to the customer and back to the depot. It then evaluates order tuples against that certificate. Let A_i be its arrival clock in absolute seconds, including departure at 28,800 s, and let R_i be its post-service return duration in seconds. Tuple j with volume q_j, service duration h_j, and window $[a_j, \ell_j]$ is admissible when

$$q_j \le Q, \qquad \max\{A_i, a_j\} \le \ell_j, \qquad \max\{A_i, a_j\} + h_j + R_i \le 86{,}400. \tag{43}$$

A bipartite matching must cover all parent customers using distinct templates within that family. The stored tuple is transferred intact, without widening or shifting its window. These tests give sufficient conditions for the chosen certificate; rejection does not prove that no alternative single-customer itinerary could serve that tuple. Feasibility-conditioned assignment can change the distribution of attributes even though each accepted tuple preserves an observed combination.

Each child view inherits its customers' assigned tuples, slices its own travel matrices, and recomputes certificates using its final station roster. Any failed time, energy, or capacity check rejects the family attempt. Under the unlimited-fleet and no-queuing assumptions, the accepted single-customer routes can jointly serve the instance. They certify existence, not route quality. The generator recomputes these certificates in its artifact checks; this is separate from the benchmark's route-set verifier.

Each view also records the fastest energy-feasible depot return from every station when departing that station fully charged. This cache can include intermediate charging and includes its time, but excludes charging at the starting station and at the depot. It is a certificate quantity, not a precomputed solver action mask.

C.6.6 Family Acceptance and Retry Scope

After depot/source/customer spatial support is frozen, materialization retries preserve the depot, structure source, customer superset, day type, and road state. They change downstream order-selection/matching and station-selection namespaces; a new tie-breaking seed need not change a greedy station roster. Up to four total materialization attempts are permitted, including the first, while nonretryable failures terminate earlier. Families that exhaust these attempts are not silently replaced.

All 7,500 released families passed their first materialization attempt and first attempted order source. Earlier construction substeps could still repeat: 21 families required an internal spatial redraw before their spatial configuration was frozen (12 used two attempts, four used three, and five used four). A family is published only after every designated view passes its checks.

C.7 Parent Families, Nested Views, and Evaluation Splits

Shared families and nested customer sets. A parent stores its selected terminals and exact pairwise travel quantities. The main training hierarchy is $1 \times \mathrm{Cus1000} \to 2 \times \mathrm{Cus500} \to 10 \times \mathrm{Cus100}$: same-level customer blocks are disjoint and their union equals the parent's customer set. Partitioning is region-first. Smaller views are therefore nested partitions of the parent, not independent rescalings of the original source template. Each validation or core test family exposes one designated chain $\mathrm{Cus100} \subset \mathrm{Cus500} \subset \mathrm{Cus1000}$.

Views with at least 500 customers inherit the parent's 50 stations. A Cus100 view greedily reselects 20 of those stations for its own customers. Its order tuples remain inherited, and its certificates are recalculated on the smaller roster. Shared terminals retain identical travel quantities, while cross-scale comparisons also reflect changes in customer composition and, for Cus100, station availability.

Geography, source separation, and day types. The ten training cities are New York, Los Angeles, Chicago, Houston, Phoenix, Philadelphia, San Antonio, San Diego, Dallas, and Fort Worth. T1 draws new families from their training communities, so individual locations may recur across training and T1 families. T2 uses held-out communities in the same cities. T3 uses the eligible pool of Jacksonville, which is held out as a whole city and contributes no training instances. All evaluation tracks use their disjoint source-day allocations.

The 500 Cus2000 parents use training communities in the ten training cities, with 50 parents per city and scalability-specific source days. Each produces one main Cus2000 view and one designated Cus1000 control. The control is one of two internal 1,000-customer blocks of that parent; only one control is released per parent. These controls are additional to the core Cus1000 views.

Weekday/weekend allocation targets 5:2 within each city and cohort, using largest-remainder integer quotas. A 500-parent allocation gives 357 weekdays and 143 weekends; a 50-parent allocation gives 36 and 14. Consequently, an aggregation of ten 50-parent city allocations gives 360/140, not an exact 5:2 ratio. All views inherit their parent's day type, speed profile, and matching source-day type.

Counting units. Table 12 distinguishes main benchmark views from paired controls. The main benchmark has 71,500 views; including 500 controls gives 72,000 views backed by 7,500 parents. Training at each of Cus100, Cus500, and Cus1000 contributes five million customer occurrences. These are repeated appearances in views, not five million distinct residential locations.

Table 12: Release counts excluding Cus50 views. Controls reuse the scalability parents and add no parent families.

Use	Parents	Cus100	Cus500	Cus1000	Cus2000	Views
Training	5,000	50,000	10,000	5,000	0	65,000
Validation	500	500	500	500	0	1,500
T1	500	500	500	500	0	1,500
T2	500	500	500	500	0	1,500
T3	500	500	500	500	0	1,500
Cus2000 main	500	0	0	0	500	500
Paired controls	0	0	0	500	0	500
Total	7,500	52,000	12,000	7,500	500	72,000

The archive also retains 101,000 Cus50 views: 100,000 training, 500 validation, and 500 T1 views. They belong to the underlying finer partition hierarchy but are excluded from the main benchmark. The physical archive therefore contains 173,000 views.

C.8 Construction Checks and Descriptive Diagnostics

Acceptance combines checks at several stages: source and terminal identities, declared spatial/source partitions, integer quota margins, nested-set sizes and membership, matrix dimensions and finite nonnegative values, inherited order tuples, charging power, and recomputed per-view feasibility certificates. No single check replaces all the others. Parent matrices are shared through per-view terminal indices, while order attributes and certificate fields are stored with the view. Identifiers and provenance fields are documented in Appendix D.2.

The release acceptance records report 7,500 verified families and 173,000 certificate-checked views, covering 24,750,000 customer-view observations including the Cus50 views. Their single-customer certificates all use zero charging visits, and the full-station return caches have no unreachable entries. These are properties of the accepted single-customer certificates; they do not imply that optimized multi-customer routes require no charging. Stage 2 uses technically verified candidate pools; candidate facilities are not manually reviewed, and travel times and energy consumption are modeled rather than field-measured.

Operational diagnostics summarize package counts, volumes, service times, window coverage and width, source-route size, radial structure, road-time neighborhoods, and community concentration. The released 106 city/scale/day-type summaries are descriptive and do not determine construction acceptance. They support examining how filtering and cross-city instantiation affect the generated corpus; they are not a statistical certification that all generated distributions or road morphologies reproduce the source operations.

D Dataset Documentation and Release

This appendix documents the composition, provenance, use conditions, and reconstruction of EVRP-TW-D. Construction procedures are specified in Appendix C, and benchmark execution in Appendix E.

D.1 Composition and Intended Use

What the release contains. The main benchmark comprises 71,500 instance views backed by 7,500 instance families across eleven U.S. cities, at workload scales of 100, 500, 1,000, and 2,000 customers. Of these, 65,000 views are training instances and 6,500 are held out for validation, the three evaluation tracks, and unseen-scale deployment. The release additionally provides 500 paired Cus1000 control views and 101,000 Cus50 views; the latter belong to the underlying partition hierarchy but are excluded from the reported benchmark. Counts by cohort are given in Table 12. The dataset archive includes the eleven city environments. Generator code, reference solver implementations, and per-instance evaluation records are released as separate benchmark artifacts (Appendix E).

Intended use. The corpus supports research on electric vehicle routing with time windows, including solver development, workload scaling, and generalization across delivery days, communities, and cities. Its instances are semi-synthetic: geography and infrastructure come from public records, while delivery requirements are transferred from observed last-mile records as joint stop-level tuples. A generated delivery day is a constructed task on real geography, not a record of a carrier's actual operations.

What the corpus does not represent. Depot candidates are drawn from mapped facility evidence: 356 of the 7,500 instance families use a group with explicit logistics or dispatch evidence, and 7,144 use a warehouse or logistics proxy whose parcel-dispatch role is unconfirmed. The depot layer is therefore not a directory of verified parcel depots. Charging powers are imputed from mode-specific national medians, capped by vehicle acceptance, and derated once to obtain battery-side effective power; they are not site-specific measurements. Reference speeds transfer national MOVES5 relationships to city roads and are not local traffic observations. Feasibility certificates establish the existence of a feasible plan, not the difficulty of finding a good one. These choices are detailed in Appendix C.

The Amazon source dataset is documented as excluding personally identifiable information, with regenerated route and package identifiers and obfuscated locations (Merchán et al., 2024). We retain source identifiers for provenance but do not transfer Amazon coordinates. Assigned delivery tuples do not represent observed deliveries to the selected service locations or their occupants.

D.2 Provenance and Identifiers

Table 7 in Appendix C.1 lists the inputs and their construction roles. Amazon records supply workload-structure templates and joint stop-level tuples of package count, volume, planned service duration, and time window. All target coordinates come from the city environments. Residential evidence uses the public NSI release and its public structure fields.

Provenance is retained across linked city, family, and view records: terminal source identifiers trace geographic inputs, while family and order-template records identify source station-days, delivery tuples, generation contracts, and seeds. View-to-family links identify shared locations and travel quantities. A reported result can thus be traced to its instance and the corresponding inputs.

Schema identifiers describe stored formats and support compatibility checks. Together with reference profiles, generation contracts, and run manifests, they identify the artifacts behind a comparison. Economic objective settings are recorded separately from physical instance parameters, so the calibration used in a run can be checked directly rather than inferred from a method name or directory.

D.3 Licensing and Attribution

The release combines inputs published under different terms (Table 13). It is organized as a collection of separately licensed components rather than under one blanket license.

For the federal sources, the release redistributes derived quantities used in construction, not the source databases themselves, and each source is cited. AFDC records are used under the AFDC data-download terms.

Geographic and delivery components. Road-network and building-derived components are derived databases of ODbL sources. They are distributed under ODbL 1.0 with attribution to OpenStreetMap contributors and to Microsoft. Amazon-derived delivery attributes are stored separately from the parent travel matrices (Appendix C.8) and are distributed under CC BY-NC 4.0 with attribution to the Amazon Last Mile Routing Research Challenge. Their transformations—stop-level aggregation, time-window intersection and clipping, workload rescaling, and omission of source coordinates—are described in Appendix C.6. The non-commercial condition applies to the Amazon-derived components and to instance views that incorporate them; it is not imposed on the ODbL components.

Table 13: Upstream sources, their published terms, and the released components derived from them.

Source	Published terms	Released components
OpenStreetMap	ODbL 1.0	Road graph, geographic evidence, and derived travel quantities
Microsoft US Building Footprints	ODbL 1.0	Building geometry for service locations
Amazon Last Mile records	CC BY-NC 4.0	Order tuples and workload-structure templates
Census TIGER/Line	U.S. government work	Boundaries and spatial units
FHWA HPMS	Public federal dataset	Road-class and speed-limit evidence
EPA MOVES5	Public EPA default database	Four reference speed factors (Table 8)
USACE NSI	Public NSI release	Residential-use and unit fields
AFDC	AFDC data-download terms	Charging-site records

Use conditions. Non-commercial use is subject to the applicable component licenses. Commercial reuse requires appropriate authorization for all restricted components involved, including project-original code and Amazon-derived delivery data. The generator accepts alternative delivery inputs, but replacing those inputs does not change the licensing terms of the generator or other components.

Software, model weights, and evaluation records. Project-original code and documentation are licensed under CC BY-NC 4.0. The same license applies to the authors' rights in supplied project-trained model weights and their original contributions to evaluation records, including route sets, metrics, and run metadata. Third-party software, input data, and embedded source-derived content retain their applicable licenses, attribution requirements, and database rights. Paper-guided reimplementations are distinguished from adapted upstream code in Appendix E. Component-level terms and attribution are provided in the supplementary package.

D.4 Maintenance and Versioning

A release-level manifest identifies the archive and code revision. City, family, and view records retain schema and provenance links; families also record their generation contract and reference profile. Schema versions describe formats and do not substitute for release identifiers or identify a particular evaluation cohort on their own.

Evaluation sets are fixed within a release. Metadata and documentation corrections are recorded separately from changes to instance content, splits, or evaluation protocols, which require a new release identifier, so that reported results remain interpretable against the release they used. The authors maintain the release. An anonymized copy accompanies this submission, and versioned artifacts will be deposited on HuggingFace with a persistent DOI upon publication.

D.5 Reconstruction and Slim Releases

Restoring derived travel caches. The reconstruction interface supports a slim instance tree that omits the four dense parent matrices while retaining family and view records, a reconstruction contract, the reference profile, an instance registry, and the charging-power median registry used by that profile. The instance registry maps views to families; restoring a view restores its shared family matrices rather than generating a new delivery day.

Restoration uses recorded road-state factors, the corresponding city graph and speed layer, and terminal-access records. Directed edge projections, along-edge offsets, and connector lengths are required: a nearest road-node identifier alone cannot describe access partway along an edge. Customers, station rosters, and delivery attributes are read from stored records rather than resampled.

Restoration checks. Restored matrices are checked for shape, data type, finite entries, and non-negativity within numerical tolerance; applicable family feasibility checks are then performed. A

complete temporary matrix directory is validated before installing a missing cache. Existing complete caches are checked and reused, while partial caches are rejected rather than overwritten. A separate verification routine compares rebuilt arrays against a reference cache under exact equality or a declared numerical tolerance. Materialization status is recorded in manifests and run reports, not established by file presence alone.

D.6 Extension Pathways

The reference task retains the assumptions of Section 3 so that established solver designs can be compared on a common basis. Relaxing an assumption is meaningful only when the data it depends on are modeled at matching fidelity: partial recharging evaluated with a linear charging curve, for example, would rank solvers on a charging trade-off that the data do not represent. Extensions of EVRP-TW-D/B are therefore organized as a dependency structure in which *data layers* of the generator must be admitted before the *benchmark tracks* that rely on them can be defined (Table 14).

Data layers. The released corpus constitutes the core layer D0 (Appendix C). Further layers differ in how far their inputs already exist. Hourly speed profiles (D1) can be derived from the MOVES5 tables already used in construction, which the release aggregates into day-type factors (Appendix C.3). Port counts and connector types (D2a) can be derived from the AFDC records already used for the power medians (Appendix C.5). Site-specific power and nonlinear charging curves (D2b), a multi-type vehicle catalog with per-type cost calibration following Appendix B (D3), road grade from elevation data (D4), package mass (D5), which the source delivery records do not provide, and depot operations such as reload and depot-charging times (D6) require additional sources.

A data layer is admitted when (i) its sources and redistribution terms are documented as in Appendix D.3; (ii) it covers every city environment of the release, or a declared subset; (iii) it is reconstructed deterministically from recorded inputs, as the travel caches are (Appendix D.5); and (iv) its derived quantities are checked against the statistics of their sources.

Table 14: Benchmark tracks, the reference assumption each relaxes, and the data layers it requires. D0: released core; D1: hourly speed profiles; D2a: charger port counts and connector types; D2b: site power and nonlinear charging curves; D3: vehicle catalog; D4: road grade; D5: package mass; D6: depot operations.

Benchmark track	Reference assumption relaxed	Required data layers
Reference task (this work)	—	D0
Partial recharging	Full recharging	D0 + D2b
Capacitated stations	No station queuing	D0 + D2a
Time-dependent travel	Static day-type travel times	D0 + D1
Heterogeneous fleet	Homogeneous vehicles	D0 + D3
Energy-realistic routing	Distance-proportional energy	D0 + D4 + D5 (optionally D1)
Multi-trip operations	Single-trip operation	D0 + D6

Unlocking a benchmark track. A track is defined only after its required layers are admitted, and it must additionally (i) state the reference assumptions it relaxes and its task semantics; (ii) extend the independent verifier of Appendix E.2 to replay every new resource or decision; (iii) regenerate feasibility certificates under the new semantics; (iv) freeze its own reference pool and reporting cohorts; and (v) reduce exactly to the reference task when the new quantities take their reference values, for example when every charging visit restores a full battery or every hourly speed equals its day-type value. The last condition keeps each track comparable with the reference task and with one another, so that extensions add evaluation settings without fragmenting the comparison protocol that Appendix G documents for prior studies. Each admitted layer or track is issued under a new release identifier (Appendix D.4); the reference task and its reported results remain fixed.

E Experimental Setup and Reporting

Section E.1 identifies each method's implementation and the changes made to it; Section E.2 specifies the route-set interface and the independent replay that accepts or rejects a returned plan; Sec-

tion E.3 defines candidate selection, the reported quantities, and the frozen reference pool; Section E.4 lists budgets, hardware, and training configurations. The task itself is specified in Appendix A and its cost calibration in Appendix B.

E.1 Method Sources and Implementation Variants

The benchmark includes an exact solver, two metaheuristics, and five learned policies. Reproducing a result requires identifying the implementation and configuration, the checkpoint for a learned policy, and the evaluation code that loads the method and processes its output. We distinguish *task adaptations*, needed to represent the reference problem, from *enhancements*, which change the search, architecture, or training procedure with the intention of improving performance. Execution and memory optimizations are documented separately. This classification does not establish an empirical benefit for each modification.

All methods use the travel conventions of Appendices A.1 and C.5: scheduling follows t^{rt}, while both the cost distance and battery consumption use d^{rt}. Training rewards and auxiliary penalties are distinct from the reported cost. The implementations below may impose additional restrictions on the routes they construct or search; these restrictions are part of their identities and are not additional assumptions of the reference task. The shared learning environment masks a station already visited by the current vehicle and resets this mask when a new route starts; this construction restriction also differs from the reference task's allowance of repeated station visits.

Implementation review. Codex assisted with paper-guided implementation, code adaptation, and training-pipeline integration. The authors checked architectures and parameter settings against the published descriptions and reviewed implementation choices where details were unspecified. Reported results characterize the implementation variants documented below. Independent replay checks solution feasibility and cost under the benchmark task; it does not establish equivalence to the original algorithm implementations.

Gurobi. Gurobi solves our directed arc-flow mixed-integer formulation defined in Appendix A.2, which represents customer visits, vehicle flow, cargo capacity, time and battery propagation, full recharging, and the reference objective. Charging stations are represented by finitely many copies. With two copies per physical station, that station can be visited at most twice across the entire solution, so an optimality certificate applies to the finite expanded model rather than to the unrestricted task. The copy count is therefore a model restriction recorded with each run.

ALNS. Our Python implementation is inspired by the partial-recharging ALNS of Keskin & Çatay (2016) and is not an import of the authors' code. It combines customer and station destroy–repair operators with adaptive operator weights and simulated-annealing acceptance. Task adaptations replace partial recharging with full recharging and introduce the reference resource matrices and cost evaluation. Additional search choices include initialization from the single-customer feasibility certificates stored with each instance, best-fit route consolidation, and scale-dependent candidate truncation. Replay caches and incremental bookkeeping reduce execution overhead.

VNS/TS. This Python implementation is inspired by the VNS/TS method of Schneider et al. (2014); the original author implementation is not publicly available. It combines neighborhood perturbations, tabu improvement, resource penalties, and charging-station moves, with the route state adapted to the reference instance quantities. Additional choices include certificate-based initialization, best-fit route consolidation with a bounded runtime, and nearest-candidate restrictions in the fast search profile, which explores a restricted neighborhood. Apply, evaluate, and rollback operations and route caches are execution optimizations.

AM-EVRPTW. This implementation adapts the released CVRP code of the Attention Model (Kool et al., 2018) to EVRP-TW, vendoring its graph encoder unchanged while the EVRP-TW decoder and training interface are local adaptations. All four policies use 128-dimensional node embeddings and a three-layer attention encoder with eight heads per layer, 512-dimensional feed-forward hidden layers, residual connections, and batch normalization. The autoregressive decoder uses an eight-head attention glimpse followed by single-head compatibility scoring with tanh clipping at 10. Task adaptations add delivery and charging attributes to node features and load, battery,

and time to the decoding context. Road quantities enter the state, transitions, feasibility masks, and cost; the encoder does not receive an explicit matrix of pairwise road relations. Training uses REINFORCE, AdamW, and a scheduled baseline. Projection caching, microbatch gradient accumulation, and distributed execution are recorded separately from the policy architecture.

TERRAN. The local implementation starts from the released EVRP-TW code of TERRAN (Tang et al., 2025), using potential-based reward shaping with PPO and the original critic architecture. Advantages use finite-trajectory Monte Carlo returns, and value fitting uses Smooth-L1 loss. Stage-specific batch sizes and learning rates are listed in Table 15; the remaining PPO settings are recorded in the released run configurations.

EVRPTW-RL. This is a paper-guided implementation of Lin et al. (2021), using Structure2Vec embeddings, context and choice attention, an LSTM decoder, and REINFORCE; author code was not used. All four policies replace sum aggregation of neighbor and time-edge messages with mean aggregation to control graph-size-dependent activation growth, which changes the network computation and is treated as an architectural modification. A station-visit auxiliary is normalized by customer count during training and does not enter reported cost. The transition from an exponential-moving-average baseline to a greedy-rollout baseline is a baseline schedule, distinct from the constraint stages of DRL-TS.

DRL-TS. Our implementation follows the two-stage design of Chen et al. (2022), based on the published description without author code. It uses node and edge graph attention and a GRU decoder, with a training schedule that moves from soft constraint penalties to hard feasibility constraints. Local changes include explicit energy-edge features and normalized, clipped violation penalties with a fixed customer-count denominator. In addition to the shared station-revisit mask, the method's action mask prohibits a charging-station action immediately after the depot or another charging station, further narrowing the routes the policy can construct. Inference enforces hard constraints.

RRNCO. This implementation adapts the released CVRP code of Son et al. (2026) to EVRP-TW, denoted RRNCO-EV in this appendix and RRNCO in the main text. It retains incoming and outgoing representations and road-relation encoding while adding electric-vehicle resource context. All four policies use the full road-relation mode. Additional changes include energy relations, distance, time, and energy biases in the decoder, a stabilized relation operator, nearest-distance relation selection, and a leave-one-out baseline. The stabilized operator changes how relations are combined and is an architectural modification. Angle relations enter the encoder rather than acting as a separate decoder bias term. Chunking and activation recomputation address memory use.

Checkpoint coverage. The evaluated configurations comprise 20 learned-policy checkpoints: Euclid-100, Road-100, Road-500, and Road-1000 for each of the five methods. Checkpoint hashes, selected epochs, saved configurations, and evaluation-code revisions identify the loaded policies.

E.2 Route-Set Interface and Independent Replay

Submitted plans. A solution record identifies its instance and implementation and stores a list of terminal sequences. Terminal 0 is the depot, terminals $1, \ldots, N$ are customers, and terminals $N+1, \ldots, N+|\mathcal{S}|$ are the physical charging stations in $\mathcal{S}$. Each submitted trip begins and ends at terminal 0; station-copy indices used inside an optimizer are mapped back to physical terminal indices before export.

Resource propagation. Replay uses the instance data rather than solver-reported arrival times, battery states, or objective values. Each trip starts at the depot opening time with a full usable battery and zero accumulated delivery volume. For every transition (i, j), replay adds t^{rt}_{ij} to the clock, subtracts $\kappa d^{\mathrm{rt}}_{ij}$ from the battery, and adds d^{rt}_{ij} to the cost distance. All used arc quantities must be finite, and battery energy may not fall below zero beyond the numerical tolerance.

At a customer, service begins at the later of arrival and the window opening; this time must not exceed the window closing. Replay then adds the customer's service duration and delivery volume. Accumulated volume may not exceed vehicle capacity. At a charging station s, arrival battery b

requires $3600 \max\{0, B - b\}/P_s^{\text{eff}}$ seconds for a full recharge, where $P_s^{\text{eff}} = \eta p_s^{\text{cap}}$ is battery-side power in kW and the derating factor is applied once (Appendix A.1); the battery is then restored to B. Return to the depot must occur within the operating horizon. Across the route set, every customer must occur exactly once.

Replay uses an absolute tolerance of 10^{-5} in the native unit of each checked resource: seconds for time, kWh for battery, and cm^3 for volume. Terminal bounds and customer occurrence counts are discrete checks. Diagnostics retain violations, recomputed cost distance, charging-visit count, and total charging time. Independence here means replay outside the solver's search or policy environment, not third-party authorship. The generation-time single-customer certificates of Appendix C.6 use a separate routine and do not replace replay of a returned plan.

Structural screens. Replay additionally rejects a trip that visits the depot internally and a trip that serves no customer. These rules apply to all solver families, and all reported results are replayed under the same verifier revision.

Dispatch and cost accounting. The deployment count used in the cost track is

$$K(x) = \sum_{(i,j)\in\mathcal{A}(x)} \mathbf{1}\{i = 0,\ j \neq 0\}. \tag{44}$$

Here $\mathcal{A}(x)$ is the multiset of traversed arc occurrences, so repeated uses of the same arc are counted separately. For separate single-trip routes, this equals the number of nonempty depot departures, including a departure to a charging station. Returning to the depot incurs distance cost but no second fixed charge; padding and depot self-loops incur no deployment charge. A learning environment's count of completed customer-serving routes is a separate diagnostic and does not replace K. Cost is recomputed from this dispatch count and the replayed cost distance using Equation (3).

Completion and candidate eligibility. For learned policies, a candidate must complete decoding successfully and pass independent resource replay. Costs of completed candidates are recomputed from exported routes; candidates are examined in nondecreasing cost and the first passing replay is retained. An incomplete rollout remains unsuccessful even if exporting its diagnostic sequence appends a depot return that happens to pass replay. Its incurred cost is diagnostic, not a feasible solution-quality observation. An internal completion flag is therefore necessary for eligibility but never sufficient to establish feasibility or determine the reported cost.

E.3 Candidate Selection, Metrics, and Reference Pools

Evaluation unit and retained solution. A configuration identifies the solver implementation and its run settings. For a learned policy, it additionally identifies the training corpus and scale, the selected checkpoint, and the decoding procedure. Let $\mathcal{I}$ be a stated evaluation cohort and $\mathcal{X}_{m,i}$ the completed candidates produced by configuration m for instance i within its declared search or decoding budget. Its retained solution is

$$x_{m,i} \in \operatorname*{arg\,min}_{x\in\mathcal{X}_{m,i}:V_{m,i}(x)=1} C(x), \tag{45}$$

where $V_{m,i}$ denotes the replay and eligibility rules of Section E.2. If the accepted candidate set is empty, $x_{m,i}$ is undefined. Exhausting a budget does not invalidate an incumbent already found within it. A proof of infeasibility, an incomplete rollout, or a diagnostic route without an accepted solution does not constitute success. For tied minima, the retained route identity records the implementation's actual choice, and all reported quality quantities refer to that same plan.

Full-cohort quantities. Let $z_{m,i} = \mathbf{1}\{x_{m,i} \text{ exists}\}$. Feasibility rate is

$$\text{FR}_m = \frac{100}{|\mathcal{I}|} \sum_{i\in\mathcal{I}} z_{m,i} \quad (\%). \tag{46}$$

Mean online wall-clock time likewise uses all instances, $\bar{t}_m = |\mathcal{I}|^{-1} \sum_{i\in\mathcal{I}} t_{m,i}$, including unsuccessful runs. Early termination contributes its recorded elapsed time, not zero and not an automatically assigned full budget. Because early failure may be fast, online time is interpreted together with FR. Offline training time is reported separately in Table 15.

Common feasible subset. For the configurations $\mathcal{M}$ participating in a comparison, define

$$\mathcal{I}_{\mathrm{cf}}(\mathcal{M}) = \{i \in \mathcal{I} : z_{m,i} = 1 \text{ for every } m \in \mathcal{M}\}, \qquad n = |\mathcal{I}_{\mathrm{cf}}(\mathcal{M})|. \tag{47}$$

Mean cost, Gap, deployment count K, and distance D use this intersection, whose size n is reported. If it is empty, these means are undefined and displayed as an em dash, not zero. No missing solution receives an artificial cost penalty. Cost, K, and D are all recomputed from the same $x_{m,i}$ rather than choosing a different candidate for each metric.

For the main-text tables, the participating quality-comparison set contains only configurations feasible on all 500 instances of the stated cohort, so their intersection is the full cohort ($n = 500$). Configurations with lower FR retain their feasibility results but have dashed quality entries; feasible-only means are not substituted. FR is always computed for every evaluated configuration on the full cohort. Any supplementary analysis using a different participating set states that set and its intersection size explicitly; means from different subsets are not directly cross-ranked.

Frozen reference pool. For each instance i, let $\mathcal{P}_i$ contain the accepted outputs from the disclosed reference configurations. The reference value and relative gap are

$$C_i^{\mathrm{BKS}} = \min_{x \in \mathcal{P}_i} C(x), \tag{48}$$

$$\mathrm{Gap}_{m,i} = 100\,\frac{C(x_{m,i}) - C_i^{\mathrm{BKS}}}{C_i^{\mathrm{BKS}}}. \tag{49}$$

The reported mean Gap is the arithmetic mean of the per-instance percentages over $\mathcal{I}_{\mathrm{cf}}$, not the percentage difference between two aggregated mean costs. Reference values use the same objective profile and are frozen for the reported comparisons. Changing a method's feasible intersection changes the averaging set, not the per-instance reference values.

The pool contains declared solver outputs and excludes both routes constructed solely to establish generation-time feasibility and the two-hour ALNS refinements of Section 6.4. The headroom experiment starts from the best of the five Road-1000 solutions per instance, a separate reference from the benchmark pool. The pool records its participating configurations, retained reference routes, and coverage for each cohort. Zero Gap denotes attainment of that reference, not certified optimality, and a later configuration can obtain a negative Gap against the frozen pool without silently updating previously reported reference values. If an instance has no reference value, its Gap is undefined; reference coverage is stated rather than replaced by a method-specific substitute or an unreported reduction of the averaging set.

E.4 Run Configurations, Budgets, and Hardware

Training schedule and checkpoint selection. All five learning-based methods follow the same curriculum. Euclid-100 and Road-100 train for 5,000 logical epochs. Road-500 initializes from the selected Road-100 checkpoint and trains for 3,000 epochs on Cus500 views; Road-1000 initializes from the selected Road-500 checkpoint and trains for 3,000 epochs on Cus1000 views. Validation runs every 100 logical epochs on 500 instances from the policy's own training domain and scale. Among checkpoints that return a verified-feasible solution on all 500 validation instances, the one with the lowest mean verified validation cost is selected. A run without any such checkpoint would be flagged as unstable; every run reported here produced at least one.

Training unit and optimizer. All policies are trained with AdamW. A logical epoch is a rollout-and-update cycle over an effective instance batch, not a pass over the training corpus. In the REINFORCE trainers it corresponds to one optimizer update after any gradient accumulation, whereas TERRAN performs multiple PPO passes. The effective instance batch is the per-GPU batch multiplied by the number of training GPUs and the accumulation factor; it excludes the number of trajectories sampled per instance. Because instance batches and trajectory counts differ across methods and scales, equal epoch counts do not imply equal sample exposure. Euclid-100 and Road-100 train on one GeForce RTX 2080 Ti (11 GiB), Road-500 on two, and Road-1000 on four.

Decoding budget and online timing. Final evaluation samples 30 trajectories per instance and retains the lowest-cost eligible candidate accepted by independent replay. There is no local search or

Table 15: Training configurations. E100 denotes Euclid-100 and Rn denotes Road-n. GPUs are GeForce RTX 2080 Ti devices. Batch is the effective instance batch per update at the end of training; LR is the initial learning rate (the actor learning rate for TERRAN). Selected is the within-stage epoch of the evaluated checkpoint. Time is the stage's wall-clock training time in hours.

Method	Train	GPUs	Batch	LR	Selected	Time (h)
AM-EVRPTW	E100	1	108	10^{-4}	3700	7.49
	R100	1	108	10^{-4}	3300	6.63
	R500	2	24	10^{-4}	3000	5.91
	R1000	4	12	10^{-4}	2300	5.83
EVRPTW-RL	E100	1	320	10^{-3}	5000	34.21
	R100	1	320	10^{-3}	4700	22.26
	R500	2	64	10^{-3}	2500	43.53
	R1000	4	48	10^{-3}	2600	84.41
DRL-TS	E100	1	44	10^{-4}	4600	8.40
	R100	1	44	10^{-4}	4900	5.62
	R500	2	4	10^{-4}	2500	14.59
	R1000	4	4	10^{-4}	2900	19.83
TERRAN	E100	1	384	10^{-4}	4700	39.64
	R100	1	384	10^{-4}	4000	37.73
	R500	2	48	10^{-4}	2800	38.27
	R1000	4	48	3×10^{-5}	2900	63.21
RRNCO	E100	1	50	10^{-4}	4800	7.31
	R100	1	50	10^{-4}	4900	5.26
	R500	2	48	10^{-4}	2700	24.99
	R1000	4	24	10^{-4}	2900	31.23

route repair; the ALNS refinement of Section 6.4 is a separate experiment. Each instance is decoded on a single GeForce RTX 2080 Ti with an instance batch of one; training with multiple GPUs does not imply distributed inference. The per-trajectory action caps are 130, 600, 1,300, and 2,500 for deployments at 100, 500, 1,000, and 2,000 customers, irrespective of training scale. Online time follows Section 5: it covers the complete 30-trajectory decoding workload for an instance, excludes instance loading and verification, and is not divided by the number of trajectories. Failed instances retain their measured time.

Cus2000 DRL-TS uses query-row chunking in its node and edge attention encoder to reduce memory use. Each query still attends to all nodes; checkpoint weights, decoder, and trajectory budget are unchanged, so chunking changes the computation only up to floating-point reduction order.

Search budgets and hardware. Baseline searches receive up to 1,800 seconds of optimization per instance, with incumbent checkpoints at 60, 300, 900, and 1,800 seconds. The retained solution is the best accepted recorded incumbent within the budget. Searches run on a host with an AMD Ryzen Threadripper 3970X (32 cores, 64 hardware threads) and 251 GiB of memory, using 30 concurrent instance workers, except Gurobi at Cus1000, which uses 15 because of memory limits. Gurobi 13.0.0 runs with one thread per instance, two copies per physical charging station, and a zero target relative MIP gap; its budget covers optimization and excludes model construction. These settings do not imply that the search reaches an optimum before the time limit. The headroom diagnostic instead allows two additional hours from its supplied starting plan.

Euclidean training corpus. All five Euclid-100 policies use the same frozen corpus of 50,000 training instances and 500 validation instances, generated with the TERRAN instance generator and a feasibility patch that rejects malformed or unreachable customer proposals and checks single-customer feasibility witnesses after canonical conversion. Each instance contains 100 customers, 20 charging stations, and one depot. Random, clustered, and mixed layouts are sampled with weights 0.3, 0.3, and 0.4; the two type-dependent resource and time-window regimes within each layout are chosen with equal probability. Depot, station, and random-customer proposals use $[0, 100]^2$ with

Table 16: Native parameters of the Euclidean corpus before canonical unit conversion. Demand is an integer from 1 to 49 divided by the listed native demand capacity. Battery is multiplied by $100/257$ to obtain kWh; service and horizon are multiplied by 60 to obtain seconds. The final column is native time per unit of native battery energy.

Type	Battery	Demand capacity	Service	Horizon	Inverse charging rate
R1	62.14	200	10	230	0.48
R2	247.66	1000	10	1000	0.12
C1	79.69	200	90	1236	3.39
C2	118.31	700	90	3390	2.28
RC1	79.69	200	10	240	0.38
RC2	273.13	1000	10	960	0.11

reachability rejection, while clustered customers use Gaussian proposals that are not clipped to that square. Mixed layouts have 50% random customers, and both split generators retain customer time windows.

We interpret one native length unit as one kilometer and one native time unit as 60 seconds. Pairwise distances are planar Euclidean and travel times use a constant speed of 60 km/h; no weekday or weekend speed calibration applies. Battery capacities are the native type-specific values multiplied by $100/257$ kWh, and edge energy is $(100/257)d_{ij}$ kWh. Native time windows, service durations, and route horizons are multiplied by 60 seconds. Customer loads retain the generator's load fractions with an equivalent cargo capacity of $18{,}500{,}000$ cm^3, and charging powers are converted to preserve native full-recharging durations: for native inverse charging rate g, power is $60(100/257)/g$ kW with charging efficiency one.

The Euclidean and Road corpora therefore differ in their resource and temporal distributions as well as in their travel representation; they are not paired instances differing only in the distance measure. Table 16 lists the native parameters before conversion. Training and validation instances are generated from separate deterministic streams, and their identifiers and contents are disjoint. Training draws follow a seeded sequence of shuffled full-pool cycles, so each instance appears once in a cycle before the pool is reused. The corpus is generated once and reused by all five methods, and its 500-instance validation set is used for checkpoint selection.

Cost coefficients. The monetary profile records a dispatch cost of 413.63 USD, electricity price 0.39 USD/kWh, and $\kappa = 100/257$ kWh/km. The distance coefficient is $0.39(100/257)$ USD/km, approximately 0.152 USD/km. Both the cost distance and battery consumption use d^{rt}, the length of the fastest-time path. Frozen instance lists, checkpoint identities, acceptance rules, and selected routes are needed together with these coefficients to reproduce a reported value.

F Supplementary Results

This appendix supplements Section 6. Quality means use all 500 instances of each cohort and are reported only when the configuration is feasible on every instance; partial feasibility of one configuration does not shrink the cohort of another. Accepted per-instance solutions from partially feasible configurations remain eligible for the reference pool (Section F.2). Search results retain the best accepted recorded incumbent within the declared budget. Costs are computed as $C = F_{\mathrm{day}}K + \alpha D$ using the full-precision coefficients in Appendix B and unrounded inputs. Reported values are rounded for display.

F.1 Search Solvers on the Direct-Deployment Cohorts

Gurobi termination status. Table 17 classifies outcomes at the 1,800-second budget. Across all 4,500 Gurobi runs on the nine T1–T3 cohorts at 100, 500, and 1,000 customers, one terminates with status `OPTIMAL` and the remainder with `TIME_LIMIT`; none reports model infeasibility. On Cus1000 T1, 498 runs have no incumbent. In the cohorts of Table 17, six runs have incumbent candidates but no accepted candidate within the budget: four at Cus500 T1 and two at Cus1000 T3. These

are verifier rejections, not proofs of model infeasibility. The observed failures therefore comprise missing incumbents and rejected candidates; they do not establish whether the finite station-copy restriction is binding.

Table 17: Gurobi outcomes within 1,800 seconds (500 runs per cohort). Feasible: at least one accepted incumbent; Other: incumbents rejected by the verifier. An earlier accepted incumbent is retained if a later one is rejected.

Split	Scale	FR (%)	Feasible	No incumbent	Model infeasible	Other
T1	Cus100	100.0	500	0	0	0
T1	Cus500	99.2	496	0	0	4
T1	Cus1000	0.4	2	498	0	0
T2	Cus1000	0.0	0	500	0	0
T3	Cus1000	15.2	76	422	0	2

Anytime behavior. Table 18 reports the best accepted cost at each recorded checkpoint. Between 900 and 1,800 seconds, ALNS lowers mean Cus1000 cost by 2.6%; at 1,800 seconds it remains 43.0% above Euclid-100 RRNCO ($5,020.36). Between 60 and 1,800 seconds, VNS/TS lowers its mean cost by 16.5% at Cus500 and 3.2% at Cus1000 while retaining large fleets (Table 2). Mean costs are still decreasing near the end of the budget. The ordering in Section 6.1 is therefore specific to the declared budget, and these checkpoints do not establish convergence.

Table 18: Mean search cost (USD) at T1 incumbent checkpoints. Each reported cost uses the same 500 instances; — means at least one instance lacks an accepted incumbent. All ALNS and VNS/TS checkpoint rows have 100% FR.

Method	60 s	300 s	900 s	1800 s
Cus100				
ALNS	489.58	463.15	455.47	451.21
VNS/TS	520.74	519.89	519.07	518.23
Gurobi	—	610.85	549.02	515.94
Gurobi FR (%)	99.4	100.0	100.0	100.0
Cus500				
ALNS	3,833.64	3,260.13	2,870.29	2,622.22
VNS/TS	106,051.45	102,792.62	96,338.90	88,533.71
Gurobi	—	—	—	—
Gurobi FR (%)	2.2	96.6	97.8	99.2
Cus1000				
ALNS	7,911.62	7,670.11	7,368.38	7,177.41
VNS/TS	298,512.97	296,871.65	293,564.93	288,943.07
Gurobi	—	—	—	—
Gurobi FR (%)	0.0	0.0	0.0	0.4

F.2 Complete Results for Road Training across Scales

Reference-pool definition. For each instance, the benchmark reference is the lowest accepted cost in the declared configuration pool. T1 Cus100/500/1000 each include the three search methods at 1,800 seconds, five Euclid-100 policies, and fifteen Road-n policies. Cus2000 includes the fifteen Road policies; Euclid-100 policies are not evaluated at this scale. Cus1000 T2/T3 each include the three search methods and five Road-1000 policies. Missing or rejected candidates contribute no reference value. Generation certificates and the two-hour refinements of Section 6.4 are excluded. Configuration identities and per-instance reference solutions are released with the frozen pool.

Gap. Table 19 reports the mean per-instance Gap over all 500 instances in each cohort against the frozen reference pool; all six cohorts have 500 reference values. The smallest mean Gaps at Cus100/500/1000/2000 are attained by DRL-TS Road-100 (1.7%), Road-500 (0.7%), Road-1000 (0.7%), and Road-1000 (0.4%), respectively. At Cus1000 across T1/T2/T3, DRL-TS Road-1000

has the smallest mean Gaps, 0.7/0.0/0.0%. Because Gap averages per-instance ratios, its ordering can differ slightly from that of mean cost.

Table 19: Mean per-instance Gap (%) to the frozen reference pool. Panel (a): T1 cohorts and the unseen-scale Cus2000 cohort; panel (b): Cus1000 across geographic splits. —: FR below 100% or not evaluated.

(a) Deployment scales					
Method	Training	Cus100	Cus500	Cus1000	Cus2000
Gurobi	—	17.4	—	—	—
ALNS	—	2.7	118.0	263.5	—
VNS/TS	—	17.8	7,957.7	15,362.0	—
AM-EVRPTW	Euclid-100	102.1	295.7	481.8	—
EVRPTW-RL	Euclid-100	56.8	225.3	355.7	—
DRL-TS	Euclid-100	20.4	112.3	163.9	—
TERRAN	Euclid-100	—	—	—	—
RRNCO	Euclid-100	84.3	97.1	157.1	—
AM-EVRPTW	Road-100	6.7	58.4	95.8	145.1
	Road-500	15.0	52.5	93.1	118.5
	Road-1000	27.9	64.3	91.1	121.7
EVRPTW-RL	Road-100	9.8	50.5	77.9	108.0
	Road-500	15.9	38.0	61.1	80.6
	Road-1000	25.8	42.0	60.0	79.7
DRL-TS	Road-100	1.7	3.6	5.0	3.8
	Road-500	1.9	0.7	1.5	0.8
	Road-1000	5.7	3.3	0.7	0.4
TERRAN	Road-100	5.4	47.0	87.9	181.1
	Road-500	17.0	13.1	58.9	72.2
	Road-1000	6.1	21.4	34.9	33.6
RRNCO	Road-100	3.7	35.3	74.0	120.5
	Road-500	7.4	12.1	25.3	45.3
	Road-1000	29.6	14.3	18.2	28.2

(b) Geographic splits at Cus1000				
Method	Training	T1	T2	T3
ALNS	—	263.5	297.3	277.1
VNS/TS	—	15,362.0	16,457.1	13,559.1
AM-EVRPTW	Road-1000	91.1	82.9	96.4
EVRPTW-RL	Road-1000	60.0	52.2	70.0
DRL-TS	Road-1000	0.7	0.0	0.0
TERRAN	Road-1000	34.9	29.1	41.5
RRNCO	Road-1000	18.2	15.2	20.8

Fleet size, distance, and time. Tables 20 and 21 complement the costs in Table 3. At Cus1000, replacing Euclid-100 with Road-100 reduces mean fleet size by 3.32–14.90 vehicles and distance by 34.7–73.5% across the four architectures with fully feasible Euclid-100 policies. The fleet term contributes 81.8–86.9% of each architecture's cost reduction, indicating that fleet consolidation dominates in this comparison. Time follows Section 5: it is the full-cohort mean online time per instance for the complete 30-candidate decoding workload, excluding instance loading and verification.

F.3 Mean-Cost Comparisons

Each comparison uses the same 500-instance cohort. Define $\Delta = 100(\bar{C}_A - \bar{C}_B)/\bar{C}_B$, negative when A is cheaper. Table 22 calculates this descriptive ratio from the mean costs reported in the main tables; it is not the mean of instance-level percentage differences. In groups (ii) and (iii), comparison configurations are selected by their observed cohort means.

At Cus100, Road-100 DRL-TS has a mean cost 1.20% below ALNS; the other four Road-100 policies have higher means. In group (ii), matching training and deployment scales gives the lowest mean cost in all 15 evaluated architecture–deployment comparisons. At Cus2000, Road-500 has the

Table 20: Mean fleet size K, distance D (km), and online time (s) for Road policies at Cus100 and Cus500. All entries have 100% FR; displayed K and D are rounded.

		Cus100			Cus500		
Method	Road-n	K	D	Time	K	D	Time
AM-EVRPTW	100	1.04	249.20	0.41	4.02	1,320.31	2.74
	500	1.13	257.62	0.44	3.88	1,241.03	2.67
	1000	1.25	294.53	0.49	4.15	1,396.47	2.81
EVRPTW-RL	100	1.07	258.13	0.62	3.83	1,288.07	12.08
	500	1.14	256.70	0.61	3.54	1,121.94	12.03
	1000	1.24	266.20	0.62	3.65	1,138.82	12.03
DRL-TS	100	1.01	190.15	0.48	2.73	648.50	3.11
	500	1.01	185.29	0.45	2.67	622.09	3.85
	1000	1.05	189.27	0.46	2.72	630.48	3.18
TERRAN	100	1.02	270.34	0.06	3.72	1,201.08	0.48
	500	1.13	293.18	0.06	2.95	809.50	0.49
	1000	1.05	215.96	0.06	3.12	864.61	0.46
RRNCO	100	1.02	214.66	0.07	3.49	1,077.19	0.79
	500	1.06	205.53	0.07	2.92	753.34	0.78
	1000	1.27	287.07	0.08	2.97	760.61	0.78

Table 21: Mean fleet size K, distance D (km), and online time (s) for Road policies at Cus1000 and Cus2000. Conventions as in Table 20.

		Cus1000			Cus2000		
Method	Road-n	K	D	Time	K	D	Time
AM-EVRPTW	100	8.25	3,006.33	7.50	18.82	7,573.69	25.26
	500	8.27	2,709.79	7.36	16.88	6,414.77	25.18
	1000	7.90	3,388.47	7.35	17.02	6,845.00	30.30
EVRPTW-RL	100	7.50	2,743.92	43.31	16.03	6,325.75	164.55
	500	6.58	3,081.60	43.31	14.03	5,134.31	163.27
	1000	6.82	2,306.71	43.34	13.98	5,078.77	163.05
DRL-TS	100	4.59	1,142.52	9.93	8.35	2,120.67	25.42
	500	4.47	1,086.42	8.04	8.14	2,014.27	25.03
	1000	4.43	1,082.64	8.41	8.10	2,023.91	28.22
TERRAN	100	7.93	2,908.01	4.34	21.41	9,324.55	15.83
	500	6.45	3,220.45	4.35	13.58	4,257.71	15.79
	1000	5.78	1,683.41	4.16	10.45	3,635.08	15.10
RRNCO	100	7.38	2,560.06	5.27	17.00	6,572.02	19.41
	500	5.42	1,514.36	5.11	11.49	3,559.56	18.90
	1000	5.12	1,374.26	5.11	10.21	2,931.99	18.77

lowest mean cost for AM-EVRPTW, while Road-1000 has the lowest mean cost for the other four architectures. These descriptive comparisons do not establish statistical significance.

Table 22: Descriptive mean-cost comparisons on 500-instance cohorts. Rn denotes Road-n. $\Delta = 100(\bar{C}_A - \bar{C}_B)/\bar{C}_B$ is calculated from the reported cohort means; negative values favor A. Cus100–1000 use T1; Cus2000 is the unseen-scale cohort.

Scale	A	B	Δ (%)
(i) Road-100 versus ALNS			
100	AM-EVRPTW R100	ALNS	3.72
100	EVRPTW-RL R100	ALNS	6.77
100	DRL-TS R100	ALNS	-1.20
100	TERRAN R100	ALNS	2.41
100	RRNCO R100	ALNS	0.72
(ii) Matched versus best non-matched Road scale			
100	AM-EVRPTW R100	AM-EVRPTW R500	-7.30
500	AM-EVRPTW R500	AM-EVRPTW R100	-3.84
1000	AM-EVRPTW R1000	AM-EVRPTW R500	-1.31
100	EVRPTW-RL R100	EVRPTW-RL R500	-5.32
500	EVRPTW-RL R500	EVRPTW-RL R1000	-2.86
1000	EVRPTW-RL R1000	EVRPTW-RL R500	-0.55
100	DRL-TS R100	DRL-TS R500	-0.20
500	DRL-TS R500	DRL-TS R1000	-1.93
1000	DRL-TS R1000	DRL-TS R500	-0.73
100	TERRAN R100	TERRAN R1000	-0.72
500	TERRAN R500	TERRAN R1000	-5.64
1000	TERRAN R1000	TERRAN R500	-16.12
100	RRNCO R100	RRNCO R500	-3.57
500	RRNCO R500	RRNCO R1000	-1.74
1000	RRNCO R1000	RRNCO R500	-5.95
(iii) Cus2000: best versus second-best Road scale			
2000	AM-EVRPTW R500	AM-EVRPTW R1000	-1.55
2000	EVRPTW-RL R1000	EVRPTW-RL R500	-0.40
2000	DRL-TS R1000	DRL-TS R500	-0.39
2000	TERRAN R1000	TERRAN R500	-22.21
2000	RRNCO R1000	RRNCO R500	-11.80

F.4 Charging Usage

Table 23 reports the mean number of charging-station visits per instance, counted from the replayed routes. Charging use separates the solution strategies compared in Section 6.1. At Cus1000, ALNS averages 0.10 visits per instance across 16.76 vehicles, so nearly all of its routes fit within one charge; VNS/TS never charges. Euclid-100 RRNCO and DRL-TS average 20.1 and 11.8 visits with about 11 vehicles, or 1.9 and 1.1 visits per vehicle: their fleet consolidation relies on en-route charging. Replacing Euclid-100 with Road-100 reduces Cus1000 charging visits by 59.5–85.1% across the four architectures with fully feasible Euclid-100 policies, in the same direction as the 34.7–73.5% distance reduction (Section F.2). DRL-TS Road-1000, the lowest-cost policy at Cus1000, still charges about 0.5 times per vehicle. These counts are descriptive and do not identify which charging decisions are necessary for feasibility.

F.5 Geographic Splits: Fleet Size, Distance, and Time

Table 24 complements Table 4. For all five Road-1000 policies, both fleet size and distance increase from T1 to T3. All five use fewer vehicles than ALNS in each split, although their travel distances need not be shorter. The fleet term accounts for approximately 80.7–90.3% of the T1-to-T3 cost increase. The splits contain different instance populations, so these changes are descriptive and do not isolate geography as a causal factor. Time follows Section 5.

Table 23: Mean charging-station visits per instance. Cus100–1000 use T1; Cus2000 is the unseen-scale cohort. Means use all 500 instances. —: FR below 100% or not evaluated.

Method	Training	Cus100	Cus500	Cus1000	Cus2000
Gurobi	—	1.300	—	—	—
ALNS	—	0.032	0.088	0.100	—
VNS/TS	—	0.000	0.000	0.000	—
AM-EVRPTW	Euclid-100	3.534	20.034	51.350	—
EVRPTW-RL	Euclid-100	0.900	10.160	25.924	—
DRL-TS	Euclid-100	1.158	5.706	11.836	—
TERRAN	Euclid-100	—	—	—	—
RRNCO	Euclid-100	5.964	9.152	20.116	—
AM-EVRPTW	Road-100	0.708	3.500	7.642	19.250
	Road-500	0.870	3.232	6.708	15.856
	Road-1000	1.310	5.020	8.842	19.112
EVRPTW-RL	Road-100	0.552	2.866	6.180	14.656
	Road-500	0.630	2.574	5.296	12.030
	Road-1000	0.570	2.592	5.312	11.980
DRL-TS	Road-100	0.932	1.966	2.756	4.378
	Road-500	0.630	1.742	2.378	3.858
	Road-1000	0.628	1.726	2.206	3.534
TERRAN	Road-100	1.120	3.952	8.878	28.516
	Road-500	1.166	4.534	8.748	19.860
	Road-1000	0.566	2.190	3.964	8.622
RRNCO	Road-100	0.842	3.012	8.146	19.354
	Road-500	0.998	1.828	3.230	8.464
	Road-1000	2.764	2.342	2.958	6.496

Table 24: Cus1000 geographic results: mean cost (USD), fleet size K, distance D (km), and online time (s). Learned rows use Road-1000; each quality mean uses 500 accepted instances. Gurobi is omitted because its FR is 0.4/0.0/15.2% on T1/T2/T3. Search methods use the 1,800-second budget. Bold marks the lowest cost per split.

Method	Cost	K	D	Time
T1				
ALNS	7,177.41	16.76	1,624.84	—
VNS/TS	288,943.07	683.34	41,456.02	—
AM-EVRPTW	3,781.91	7.90	3,388.47	7.35
EVRPTW-RL	3,171.85	6.82	2,306.71	43.34
DRL-TS	**1,997.51**	4.43	1,082.64	8.41
TERRAN	2,647.91	5.78	1,683.41	4.16
RRNCO	2,325.52	5.12	1,374.26	5.11
T2				
ALNS	7,815.28	18.29	1,641.53	—
VNS/TS	320,418.39	758.00	45,361.38	—
AM-EVRPTW	3,599.54	7.73	2,639.21	7.82
EVRPTW-RL	2,996.83	6.49	2,058.31	43.60
DRL-TS	**1,969.09**	4.40	982.55	8.43
TERRAN	2,536.03	5.58	1,513.05	4.20
RRNCO	2,262.48	5.01	1,242.34	5.15
T3				
ALNS	8,756.74	20.41	2,083.26	—
VNS/TS	309,771.09	733.21	42,774.84	—
AM-EVRPTW	4,591.52	9.67	3,904.50	7.46
EVRPTW-RL	3,988.55	8.42	3,343.74	44.00
DRL-TS	**2,344.66**	5.17	1,358.64	8.37
TERRAN	3,299.72	7.11	2,358.88	4.12
RRNCO	2,821.35	6.14	1,845.05	5.03

F.6 Headroom Diagnostic Details

Refinement setup. For each Cus1000 T1 instance, the warm start is the lowest-cost verified solution among the five Road-1000 configurations, one per architecture; this selection pool is distinct from the benchmark reference pool. DRL-TS Road-1000 supplies the warm start on all 500 instances, with mean cost USD 1,997.51. ALNS is run in two disjoint batches of 250 instances, with 30 worker processes per batch and base seed 2026; per-instance seeds are retained in the release. Each instance receives a continuous 7,200-second search budget, excluding solver construction, and incumbents are saved every 900 seconds. The output retains the best accepted incumbent, so no reported cost increases. All 500 refined solutions are accepted by the independent verifier of Appendix E.2. The refined solutions are excluded from the benchmark reference pool.

Refinement statistics. After 7,200 seconds of warm-start ALNS, costs decrease on 437 of the 500 Cus1000 T1 instances (87.4%) and remain unchanged on 63 (12.6%). Mean cost falls from USD 1,997.51 to 1,990.93, a 0.329% reduction in the cohort mean. The median per-instance cost reduction is 0.194%, with an interquartile range of 0.053–0.551%. Mean distance falls from 1,082.64 to 1,039.25 km; the median signed distance change is -2.334%. Fleet size remains unchanged on every instance, with a mean of 4.432 vehicles, so all observed savings come from the distance term.

Per-instance cost reduction and signed distance change are defined as $100(C_0 - C_1)/C_0$ and $100(D_1 - D_0)/D_0$, respectively. Quantiles use all 500 instances, including zero improvements, with linear interpolation. These improvements are specific to the tested refinement procedure and budget. An unchanged fleet is not a certificate of fleet minimality, and the results do not establish cost optimality.

G Prior Evaluation Protocols

This appendix records the conditions behind results reported in seven selected EVRP-TW studies. Three kinds of information are distinguished: the optimization problem, the implementation, and the evaluation protocol. Table 25(a) records objectives and resource models; panel (b) records implementation availability, budgets, statistical sets, references, and success criteria. V1–V7 identify the inspected source versions and qualifications in Section G.3. The final row of each panel lists the corresponding settings of EVRP-TW-B for comparison. Our own solver implementations and their adaptations are documented separately in Appendix E.

G.1 Modeling Assumptions

The EVRP-TW name covers objectives that rank feasible plans differently and resource models that admit different plans. A fleet-first lexicographic objective differs from an additive vehicle-acquisition cost. Full and partial recharging permit different charging decisions, and finite station capacity couples the schedules of different vehicles. Charging policy alone does not specify charging duration: the inspected Chen formulation uses a fixed station service time followed by a battery reset, whereas other formulations relate duration to the energy restored. Panel (a) therefore separates the charging decision from the charging-time model, and distinguishes an optimization objective from additional penalties used during learning.

G.2 Implementation and Evaluation Protocols

Implementation availability was determined as of 17 September 2026 by inspecting each source version, its artifact statement, and any author repository identified from it. **P** denotes an identified, accessible public author repository; **AR** denotes an explicit statement that source code is available on request; **NI** denotes that an author solver release was not identified in those sources. NI reflects what that inspection found and does not establish that code never existed or is unavailable elsewhere. Released instances, solution files, and a stated programming language are distinct from a solver source release, and current access to a repository does not identify the revision used in the published experiments. TERRAN reports reconstructing VNS/TS from its published description because the original implementation was not publicly released (Tang et al., 2025); a method name alone therefore does not identify the code used in a comparison.

Table 25: Modeling assumptions and evaluation protocols in seven selected EVRP-TW studies and in EVRP-TW-B: (a) optimization and resource models; (b) implementation and evaluation. V1–V7 identify the source qualifications in Appendix G.3.

(a) Modeling assumptions

Study	Objective	Charging policy	Fleet	Energy model	Charging model
Schneider et al. (2014) [V1]	Fleet size, then distance.	Full; station revisits.	Homogeneous; size optimized.	Constant rate times distance.	Uniform constant rate; duration depends on energy restored.
Keskin–Çatay (2016) [V2]	Fleet size, then distance; displayed MILP minimizes distance.	Partial; full-charge and fixed-fraction comparisons.	Homogeneous; size optimized.	Constant rate times distance.	Linear in energy added; full initial battery.
Hiermann et al. (2016) [V3]	Vehicle-type acquisition cost + distance.	Full; station revisits.	Heterogeneous EVs; size/mix selected; no count cap in shown model.	Type-dependent constant rate; shared rate in tests.	Linear in energy added; shared charging rate in tests.
Lin et al. (2021) [V4]	Euclidean distance; training adds fleet, station, and battery penalties.	Full at each visit.	Fixed-size homogeneous fleet; scenario-specific counts.	General $f(i, j)$; normalized distance-linked test setting.	Linear full-recharge duration; normalized experimental rates.
Chen et al. (2022) [V5]	Asymmetric distance; soft-stage violation penalties.	Full; revisits allowed; depot/CS-to-CS masked.	Homogeneous; count cap unspecified.	Constant λ times distance.	Fixed station time s_i, then full battery reset; no explicit arrival-SOC duration.
Lam et al. (2022) [V6]	Sum of arc distances.	Partial; separate full-charge variant.	Homogeneous; at most K; large K permits unlimited.	Distance-proportional.	Linear/three-segment PWL; 1, 2, or unlimited chargers; nonpreemptive; no consecutive CS visits.
TERRAN (2025) [V7]	Total Euclidean distance.	Mandatory full charge.	Homogeneous; unlimited.	Distance-proportional; constant speed.	Constant charging rate; no station queue.
EVRP-TW-B (Ours)	Daily vehicle/driver charge + energy cost.	Full; station revisits.	Homogeneous; size optimized; no count cap.	Constant rate times road distance.	Constant effective power; duration depends on energy restored; no station queue.

"Then" denotes lexicographic optimization. CS: charging station; SOC: state of charge; PWL: piecewise linear. Objectives and training penalties are distinguished. An unspecified fleet cap is not an assertion of an unlimited fleet. Rows V1–V7 describe the prior studies, not our adapted solvers.

Table 25: Modeling assumptions and evaluation protocols in seven selected EVRP-TW studies and in EVRP-TW-B (continued).

(b) Implementation and evaluation protocols

Study	Code	Compute/decoding budget	Statistical set	Reference	Success meaning
Schneider et al. (2014) [V1]	NI	TR2012: add a vehicle after 500 unsuccessful feasibility iterations; 200 distance iterations; 100 TS/VNS step. Small CPLEX: 7,200 s.	36 small; 56 C100; best of 10 heuristic runs.	Small: optimum/incumbent. Large: test-wide best; fleet/distance separate.	Feasibility, matching an incumbent, and proving optimality differ.
Keskin–Çatay (2016) [V2]	NI	ALNS: 25,000 iterations; small CPLEX: 7,200 s.	36 small; 56 large. Fixed-setting means and study-wide best; ten runs explicit only for tuning.	Full-charge BKS; partial-charge CPLEX. Distance gaps condition on equal fleet sizes.	Feasibility/BKS match/proof differ. CPLEX time is termination time.
Hiermann et al. (2016) [V3]	NI	ALNS: 2,000; variants 800 or at most 1,500 with 800 nonimproving iterations. BnP: 2 h small; 8 h large.	10 heuristic runs; best/mean; cost sets A/B/C. Appendix BnP tables condition on obtaining an LB.	BnP/heuristic BKS; only marked results proved optimal.	LB, feasible UB, and proof are separate states.
Lin et al. (2021) [V4]	NI	Greedy; best-of-100; beam 3. Main-table reporting limit: 15 min/case. Separate 2 h throughput test.	100 instances per scenario; seven scenarios, $N = 5$–100; mean distance/time.	Best compared method; gap aggregation unspecified.	Demand completion; independent replay and failed-run denominator unspecified.
Chen et al. (2022) [V5]	AR	Greedy/best-of-1,280; SCIP: 180 s; SA-VNS: $1{,}000 \times 10$ iterations.	1,000 each at C10/20/50/100; mean objective/time; 1% rank-sum test.	$100(O_m - O_b)/O_b$; O_b: best compared method, not necessarily a certified optimum.	SCIP: all C10 and 400 C20 proved optimal; no C50/100 proof. Neural replay/failure denominator unspecified.
Lam et al. (2022) [V6]	NI	1 h and 16 GB/case; single thread.	Two sets of 224; 56 at each $N = 25, 50, 75, 100$. Table 2 conditions on Benders-cut activity.	Primal–dual gap; denominator unspecified; missing bound: 100%.	Solved = proved optimal or infeasible. Timeout time: 3,600 s.
TERRAN (2025) [V7]	P†	Best-of-100; beam 20. Large CPLEX: 1 h. Small-instance time cap and reported times not reconciled in the paper.	56 C100; small-table aggregation and repeat count unspecified; best and stochastic mean/SD.	CPLEX/method comparisons; no frozen common BKS specified.	Feasible complete routes; independent physical checker unspecified.
EVRP-TW-B (Ours)	P	Search: 1,800 s per instance; learned policies: best-of-30.	500 instances per cohort; FR over the full cohort; quality means over the common feasible subset.	Frozen pool of declared solver outputs; per-instance Gap.	Plan accepted by an independent verifier.

P: identified public author code; AR: explicitly available on request; NI: no author solver release identified in the inspected sources. Availability is as of 17 September 2026 and is not an exhaustive code search. LB/UB: lower/upper bound; BnP: branch-and-price. †TERRAN reports reconstructing its VNS/TS baseline because the original implementation was not publicly released. V1–V7 state the source-version qualifications. Rows V1–V7 describe the published methods, not our adapted implementations. Budgets in different units are not ranked.

Panel (b) keeps statistical sets, reference values, and success criteria separate because each changes the interpretation of a reported number. Keskin's distance comparisons condition on equal fleet sizes; Hiermann's selected appendix tables condition on obtaining a lower bound; and Lam counts both proved optimality and proved infeasibility as solved. These quantities cannot be read as a common feasible-plan success rate. Lin and Chen use best-compared-method references, Hiermann combines branch-and-price and heuristic results in a BKS, and Lam reports a primal–dual gap. A zero gap to an incumbent or BKS does not certify optimality. An unstated gap denominator or aggregation formula remains unspecified.

Budgets must also be interpreted in their stated units. A fixed iteration count, a per-instance time limit, best-of-k decoding, and a batch throughput window measure different allowances. Decoding candidates are not independent training runs, and a reported stochastic standard deviation does not determine the number of training repeats. The inspected descriptions do not establish a common separation of loading, preprocessing, model construction, search or decoding, and verification time. Missing timer components and failure denominators are not inferred from implementation conventions.

G.3 Source Versions and Qualifications

V1: Schneider. The numerical protocol was checked in Technical Report 02/2012 (Schneider et al., 2012), not against the 2014 version of record. Its iteration limits are therefore explicitly attributed to the report. The inspected locations are Section 3 and Sections 5.1–5.2, including Tables 2–4.

V2: Keskin. The inspected source is Chapter 2 of the author's 2018 thesis (Keskin Özel, 2018), which explicitly attributes the study to the 2016 article (Keskin & Çatay, 2016). This source is not treated as verification of the publisher's version. Ten repetitions are explicit for parameter tuning, but are not established for every main result table.

V3: Hiermann. The source is the peer-reviewed author manuscript (Hiermann et al., 2016). Its appendix branch-and-price tables retain cases that yield a lower bound within eight hours; this selected set is not an all-instance success denominator.

V4: Lin. The accepted manuscript (Lin et al., 2021) selects a beam output by final probability, while sampling selects by distance. The two-hour experiment measures throughput, not a two-hour allowance for each instance. Neither 100 decoding candidates nor reported averages determine an independent-training repeat count.

V5: Chen. The inspected publisher chapter (Chen et al., 2022) (pages 356–368) contains the methods, results, and conclusion. The original full-charge update uses station service time s_i, without an explicit duration determined by arrival state of charge. Independent training repeats and detailed timer boundaries are not specified in the chapter.

V6: Lam. The author manuscript (Lam et al., 2022) does not specify the experimental fleet-limit value or repeated-run count in its main text. Table 1 covers all cases, whereas Table 2 conditions on Benders-cut activity.

V7: TERRAN. Section V.A of the inspected author manuscript (Tang et al., 2025) specifies 500,000 training steps for the 5- and 15-customer models and five million for the 100-customer model; a single five-million-step budget therefore does not describe all scales. The general small-instance test-set description and the explicit twelve-instance C15 analyses have different scopes; the latter does not establish every small-table denominator. Appendix C states a one-minute small-instance limit, whereas Table II reports 148 s for CPLEX and 63 s for VNS/TS at five customers. The inspected text does not explain how these times relate to that limit, so their timing composition remains unspecified. The stochastic repeat count is also unspecified.

This comparison documents specific differences across studies; it does not rank them or invalidate comparisons made under their own stated settings. Our own training configurations, timing boundaries, evaluation cohorts, and reference pools are documented in Appendix E.