Carbon-Aware Offline Reinforcement Learning for Joint Inventory Replenishment and Transportation Scheduling in Multi-Echelon Supply Chains

Pan Li [1], Menglin Bian [2], Yizhen Lin [3]

[1] University of Hull, UK

[2] University of Missouri, USA

[3] Stanford University, USA

* Corresponding Author Email: menglinbian@ieee.org

Abstract. Joint replenishment and transportation decisions couple service, inventory, vehicle utilization, and carbon emissions over time. Offline reinforcement learning (RL) is attractive when operational logs exist but online exploration is unsafe; however, distribution shift and cumulative carbon constraints make direct offline Q-learning unreliable. This paper proposes budget-conditioned carbon-aware conservative Q-learning (CA-CQL), a discrete-action offline method that combines a remaining-budget state, carbon-shaped conservative value learning, a learned behavior-support prior, a state-dependent carbon shadow price, and a pathwise carbon-feasibility screen. A reproducible supplier–distribution-center–three-retailer benchmark is constructed because public sales datasets lack synchronized replenishment actions, vehicle schedules, loads, and emissions. Transport emissions are calibrated with the official UK Government 2026 vehicle-kilometer conversion factors. Across five independently generated offline datasets and 250 evaluation episodes per policy, CA-CQL achieves an operating cost of 16,865 ± 404, emissions of 14,119 ± 57 kg CO_2e, a 92.18 ± 0.88% fill rate, and zero carbon-cap violations. Relative to a support-regularized CQL baseline, it reduces emissions by 110.8 kg and the violation rate by 13.6 percentage points, with a 1.36% cost increase and a 0.63-point fill-rate decrease. Ablations show that the behavior prior prevents severe under-replenishment and that the carbon screen is necessary for hard pathwise compliance. The results establish a transparent compliance–service trade-off rather than claiming universal dominance.

Keywords: Offline reinforcement learning, conservative Q-learning, multi-echelon inventory, transportation scheduling, carbon budget, supply-chain control.

1. Introduction

Multi-echelon supply chains coordinate inventory positions at upstream and downstream facilities while selecting when, how much, and how fast to ship. The classical multi-echelon inventory literature establishes the value of coordinated replenishment [1], while inventory-routing models integrate delivery timing, quantity, and routing under deterministic or stochastic demand [2-6]. Carbon regulation adds another intertemporal coupling: a dispatch that improves today's service consumes part of a finite emissions budget and can restrict tomorrow's feasible transportation choices [7-12].

Reinforcement learning offers a flexible way to optimize sequential policies when closed-form dynamic programs are intractable. Recent work has demonstrated deep RL for inventory control and logistics [24-28]. Yet most studies train through repeated simulator interaction. In an operational setting, exploratory stockouts, excess orders, or unnecessary express trips are costly and potentially noncompliant. Offline RL instead learns from a fixed historical log [16-23], but naive bootstrapping can assign high values to poorly covered actions. This failure is especially damaging in joint inventory–transport action spaces, where the Cartesian product creates many feasible but weakly observed combinations.

A second gap is that expected carbon penalties do not guarantee episode-level compliance. A policy can achieve a low mean footprint while occasionally exceeding a contractual or internal cap.

Safe RL formulates constraints through constrained Markov decision processes (CMDPs) [13], [15], and recent constrained offline methods address fixed-data safety [21], [22]. However, their generic formulations do not exploit a useful property of logistics emissions: dispatch emissions are nonnegative and can be calculated before an action is executed. This permits a simple pathwise feasibility projection that is unavailable for many delayed safety costs.

This work studies a supplier–distribution-center–retailer network with stochastic demand, finite capacities, lead times, consolidated regular transport, direct express transport, and a finite horizon carbon cap. The contribution is fourfold. First, the problem is formulated as a budget-augmented offline CMDP with a 159-action joint replenishment–transport space. Second, CA-CQL combines conservative value learning with an explicit behavior-support prior and carbon-conditioned deployment scoring. Third, a one-step emissions screen provides a pathwise cap guarantee without online learning or future-demand information. Fourth, a fully reproducible benchmark, fixed offline datasets, model checkpoint, official emission-factor workbook, raw results, ablations, sensitivity tests, and reference-verification file are released with the paper.

The study deliberately does not present synthetic observations as an enterprise case. Public retail datasets such as those used in the M5 competition contain demand histories [29], but they do not contain the historical action, vehicle, load, route, and emissions fields required to identify an offline joint control policy. Consequently, stochastic demand and operating parameters are synthetic and transparent, whereas vehicle emission coefficients are taken from the official 2026 UK conversion-factor workbook [12].

2. Related Work

2.1 Joint Inventory and Transportation Control

The structural foundation of multi-echelon inventory control dates to Clark and Scarf [1]. Inventory-routing research subsequently treated the coordination of customer inventory and distribution decisions [2], including stochastic consumption and route failures [3], direct-delivery Markov decision formulations [4], approximate dynamic programming [5], and decomposition methods [6]. Carbon-aware extensions typically place taxes, caps, offsets, or trading rules inside mathematical programs [7]. Pollution-routing work models the dependence of fuel and emissions on vehicle operation [8], and surveys emphasize the importance of load-sensitive, activity-based emissions accounting [9]. These methods are powerful when forecasts, distributions, or planning models are known, but they do not directly solve policy learning from heterogeneous historical decisions.

2.2 Offline and Constrained Reinforcement Learning

Offline RL learns without further environment interaction. Batch-constrained Q-learning limits actions to data-supported regions [16], BEAR analyzes bootstrapping-error accumulation [17], CQL lowers values for out-of-distribution actions [18], and IQL avoids explicitly evaluating unseen actions [20]. D4RL provides standard fixed-data benchmarks [19], while broader reviews identify reliable policy selection and distribution shift as unresolved practical issues [23]. For constraints, CMDPs formalize cumulative cost limits [13], CPO enforces constraints in online policy optimization [15], and COPO and CPQ extend safety ideas to offline data [21], [22].

CA-CQL is not proposed as a general replacement for those methods. It is a domain-specific synthesis for discrete logistics control. Its behavior prior is a support regularizer rather than a hard imitation constraint; its carbon signal is embedded both during training and action selection; and its final screen exploits deterministic pre-dispatch emissions. The screen separates hard feasibility from soft economic trade-offs, allowing the learned score to choose among cap-feasible actions.

2.3 Reinforcement Learning in Supply Chains

Surveys of logistics and supply-chain RL report rapid growth but also limited use of industrially complete datasets [24], [25]. Deep RL has been evaluated for multi-echelon inventory systems [26], multi-product multi-node capacity-constrained networks [27], and canonical lost-sales, dual-sourcing, and multi-echelon problems [28]. These studies motivate neural policies for high-dimensional operations. The present work differs by focusing on fixed-data learning, joint transportation mode/schedule decisions, explicit vehicle emissions, and hard episode-level carbon compliance.

Beyond supply chains, DRL-Adapt demonstrates that a carefully structured dueling deep Q-network with prioritized replay can jointly optimize convergence speed, control overhead, and stability in large-scale network control [30]. Its strong cross-scale generalization provides encouraging evidence that domain-informed state, action, and reward design can yield practical adaptive policies in complex operational systems.

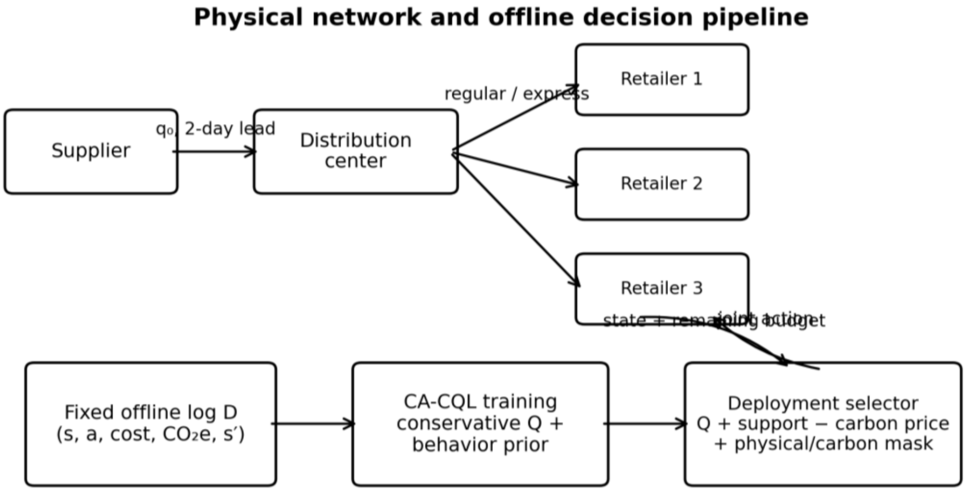

Figure 1. Physical network and the fixed-data CA-CQL pipeline

3. Problem Formulation

3.1 Network, State, and Joint Action

The physical network contains one external supplier, one distribution center (DC), and three retailers. The DC orders from the supplier with a two-day lead time. Retailers receive either a consolidated regular shipment with a two-day lead time or direct express shipments with a one-day lead time. Demand is observed after dispatch and follows an overdispersed seasonal count process. Negative retailer inventory represents backlog; the fill rate is fulfilled demand divided by total demand.

At day t, the 19-dimensional state s_t contains normalized DC inventory, three retailer net inventories, two upstream pipeline positions, six retailer pipeline positions, three seasonal demand means, sine and cosine calendar encodings, the normalized remaining carbon budget $z_t=(B-C_t)/B$, and progress $\tau_t=t/H$. The horizon is H=60 days and the nominal cap is B=15,000 kg CO_2e.

The joint action is $a_t=(q_t^0, q_t^1, q_t^2, q_t^3, m_t)$. Upstream order q_t^0 is selected from {0, 20, 40}; each retailer allocation q_t^i is selected from {0, 5, 10}; and m_t is consolidated regular or direct express. Removing redundant express actions with zero downstream shipment yields 159 discrete actions. A physical feasibility mask enforces DC availability, transport capacity, and an upstream inventory-position bound.

3.2 Cost, Emissions, and CMDP Objective

The operating cost combines DC and retailer holding cost, backlog cost, order/dispatch setup costs, and distance-based transport cost. The immediate RL reward is $r_t=-cost_t/100$. Transportation emissions $e(a_t)$ are calculated before execution. For an articulated heavy-goods vehicle, the kg CO_2e/km factor is linearly interpolated between the 0%-laden and 100%-laden endpoints; the same calculation uses rigid-truck endpoints for express trips. The exact factors are 0.65316 and 1.07508 kg CO_2e/km for average non-refrigerated articulated HGVs, and 0.46656 and 0.54675 kg CO_2e/km for rigid >3.5–7.5 t HGVs [12].

TEAS shows that replacing coarse request counts with workload-aligned effective-token signals and explicitly accounting for energy asymmetry can markedly improve service-cost control in dynamic infrastructure [31]. Following the same domain-aligned principle, our freight model evaluates load-sensitive kg CO_2e for each candidate action instead of treating all dispatches as equivalent.

$$\min_\pi \quad E_\pi\left[\sum_{t=0}^{H-1} \gamma^t \, cost_t \right] \quad \text{subject to} \quad \sum_{t=0}^{H-1} e(a_t) \le B \tag{1}$$

Equation (1) uses a pathwise finite-horizon carbon cap rather than only an expected discounted constraint. This matches an operational interpretation in which each 60-day planning episode must remain within a declared transport budget. The dataset D contains transitions (s,a,r,e,s',d) generated by historical behavior and no interaction is permitted during policy training.

Table 1. Nominal environment and training configuration

Parameter	Value
Horizon / carbon cap	60 days / 15,000 kg CO_2e
Demand means	3, 4, 5 units/day; seasonal overdispersion
DC / retailer capacity	120 / 50 units
Regular / express capacity	30 / 20 units
Lead times	Upstream 2; regular 2; express 1 day
Action count / state dimension	159 / 19
Offline data per seed	220 episodes = 13,200 transitions
Behavior mixture	68% heuristic; 17% low-carbon; 15% random feasible
Network / optimizer	2×128 SiLU Q-net; AdamW, 3×10^{-4}
RL settings	γ=0.99; CQL α=0.25; batch=256; 1,600 updates
Evaluation	5 seeds × 50 common-random-number episodes

4. Budget-Conditioned Ca-Cql

4.1 Conservative Value Learning with a Behavior Prior

A behavior network $\mu_\varphi(a|s)$ is first fitted by masked cross-entropy on D. It estimates relative action support among physically feasible actions. The Q-network and target network use two 128-unit hidden layers. For a logged transition, carbon is normalized by B and the training shadow price increases when cumulative spending is ahead of elapsed time: $\lambda_{train}(s)=6+20\cdot\max(0, (1-z)-\tau)$. The shaped reward is $\tilde{r}=r-\lambda_{train}\, e/B$.

$$L_Q = E_D[(Q_\theta(s,a)-y)^2] + \alpha\, E_s\left[\log \sum_{a'\in A_f(s)} \exp Q_\theta(s,a') - Q_\theta(s,a)\right] \tag{2}$$

$$y = \tilde{r} + \gamma\, Q_{\bar{\theta}}\left(s', \operatorname{argmax}_{a'\in A_f(s')} [Q_{\bar{\theta}}(s',a')+0.8 \log \mu_\varphi(a'|s')]\right) \tag{3}$$

The first term in (2) is a smooth-L1 Bellman loss in implementation, and the second is the discrete CQL penalty over physically feasible actions $A_f(s)$. Equation (3) uses the behavior prior in target action selection. This does not force behavior cloning: high-value actions can depart from the modal behavior, but unsupported combinations receive lower deployment and backup scores.

The hierarchical IC-layout framework of Zhang et al. offers an instructive example of how congestion-aware reward shaping and adaptive action-space pruning can make high-dimensional, multi-objective RL more efficient and feasible [32]. Consistent with this constraint-aware design philosophy, CA-CQL applies physical-feasibility masks in both conservative backups and deployment scoring, concentrating learning on admissible replenishment-transport combinations.

4.2 Carbon-Conditioned Deployment Score

At deployment, the learned networks are frozen. CA-CQL ranks each currently feasible action with

$$\mathrm{Score}(s,a)=Q\_\theta(s,a)+\beta \log \mu\_\varphi(a|s)-\lambda\_\mathrm{dep}(s)e(a)/B \quad (4)$$

$$\lambda\_\mathrm{dep}(s)=18+120\cdot\max(0,(1-z)-\tau)+120\cdot\max(0,0.20-z) \quad (5)$$

Where $\beta=1.2$. The first adaptive term responds when budget use is ahead of schedule, and the second increases scarcity pressure when less than 20% of the cap remains. This state-dependent score supports zero-shot use under alternative caps because z is recomputed from the deployment cap. The policy does not update its weights and does not observe future demand.

4.3 Pathwise Carbon-Feasibility Screen

Before the argmax in (4), the mask removes any action whose known immediate emissions exceed B-C_t. Because $e(a)\geq 0$, this gives a simple induction guarantee: if C_0=0 and e(a_t)≤B-C_t at every step, then C_{t+1}=C_t+e(a_t)≤B; therefore C_H≤B. The zero-shipment action remains available, so the masked action set is never empty. This guarantee concerns the modeled transport emissions and cap; it does not imply lifecycle carbon neutrality.

4.4 Computational Procedure

For each seed, the workflow is: (1) generate and save one fixed offline dataset; (2) train behavior cloning, offline DQN, vanilla CQL, support-CQL, and CA-CQL without simulator interaction; (3) evaluate learned and heuristic policies on identical demand seeds; (4) evaluate deployment ablations and carbon-cap sensitivity; and (5) write all seed-level and aggregate outputs. The delivered code contains no hard-coded result table. Training five seeds on CPU in the supplied environment required approximately 182 seconds for the reported configuration.

Table 2. Ca-cql training and carbon-feasible deployment

Step	Operation
1	Input the fixed log D, action-emission map e(a), physical masks, horizon H, and cap B.
2	Fit $\mu\varphi(a\|s)$ by masked cross-entropy on logged actions; initialize $Q\theta$ and target $Q\bar{\theta}$.
3	For each minibatch, compute budget pace, λtrain, and shaped reward $\tilde{r}=r-\lambda\mathrm{train}\ e/B$.
4	Update $Q\theta$ with the Bellman loss plus the feasible-action CQL penalty in (2); softly update $Q\bar{\theta}$.
5	Freeze all weights. At state s, form the physical mask and remove actions with $e(a)>B-Ct$.
6	Select the remaining action maximizing (4), execute it, and update $Ct+1=Ct+e(a)$.

The deployment calculation is linear in the number of enumerated actions: the networks produce one vector of 159 scores and masking is $O(|A|)$. This is negligible for the present action set. For larger product-location spaces, the same principles can be retained with factorized action heads, hierarchical shipment construction, or candidate generation, while the final emissions screen remains an inexpensive deterministic filter.

5. Experimental Design

5.1 Synthetic Benchmark and Data Provenance

The demand generator is a Gamma–Poisson mixture with retailer means (3, 4, 5), weekly seasonality, a mild horizon trend, and a 6% probability of a common demand shock. The design produces overdispersion and intermittent coordination stress without asserting that it represents a specific firm. Each seed has 13,200 transitions; the five datasets contain 66,000 transitions in total. On average, the logs cover 147 of 159 actions (92.45%). The data-collection mixture includes a standard base-stock-like heuristic, a consolidation-oriented low-carbon heuristic, and random feasible actions.

The emissions workbook is the official UK Government 2026 flat-format file. The audit extract in the results package records the exact rows used. These factors are activity-based vehicle-kilometer factors and align with the broader accounting emphasis of the GHG Protocol and GLEC framework [10-12]. Distances, capacities, costs, and demand parameters remain scenario assumptions and are listed in the code and configuration JSON.

Table 3. Fixed offline dataset diagnostics

Seed	Transitions	Unique actions	Coverage %	Mean kg/step
0	13,200	147 / 159	92.45	235.43
1	13,200	147 / 159	92.45	234.67
2	13,200	147 / 159	92.45	235.65
3	13,200	147 / 159	92.45	236.06
4	13,200	147 / 159	92.45	236.17

Table 3 is computed directly from the saved NPZ logs. All five datasets have the same transition count and the same 147-action support, while their demand realizations, rewards, and carbon usage differ. The twelve absent actions are not deleted from the learner: they remain in the output layer and are therefore a direct test of whether conservative learning and the behavior prior suppress unsupported joint combinations.

5.2 Baselines and Metrics

Baselines are: a standard heuristic; a low-carbon consolidation heuristic; behavior cloning (BC); offline DQN; vanilla discrete CQL; and Support-CQL, which uses the same CQL value architecture and behavior prior as CA-CQL but has no carbon shaping, carbon price, or carbon screen. Support-CQL is the primary learning baseline because it isolates the carbon-specific components while retaining protection against unsupported actions.

Metrics are total operating cost, transport kg CO_2e, fill rate, and the fraction of evaluation episodes exceeding the 15,000 kg cap. Each method is evaluated on 50 episodes for each of five learning seeds. Tables report the mean and standard deviation across the five seed-level means. Paired comparisons use common random-number evaluation seeds. Given only five independent training datasets, both paired t intervals and exact two-sided sign-flip tests are reported; the latter has a minimum attainable nonzero p-value of 0.0625 when all five differences share one sign.

5.3 Ablations and Budget Sensitivity

Deployment ablations remove the behavior prior, replace the adaptive overspending term with a fixed carbon price, or remove the final carbon screen. They use separately seeded evaluation episodes to avoid reusing the main table. For budget sensitivity, the same trained policy is evaluated without retraining under caps of 13,500, 15,000, and 16,500 kg CO_2e; the normalized budget state and screen use the new cap.

5.4 Implementation and Statistical Protocol

All neural methods use the same normalized 19-dimensional observations and the same enumerated action index. Neural baselines are trained for 1,600 minibatch updates with batch size 256, AdamW learning rate 3×10^{-4}, discount 0.99, and soft target updates. The behavior prior and Q-networks are trained only on the saved dataset for their seed. No evaluation trajectory is inserted into a replay buffer, and no model selection is performed on the test episodes. The seed-0 CA-CQL state dictionary is included to permit a direct policy audit.

For every learning seed, the competing policies face identical evaluation demand streams, which reduces comparison noise without making the five training datasets dependent. Aggregation treats the seed-level mean as the independent unit, rather than incorrectly treating all 250 episodes as independent training replications. The exact sign-flip test enumerates all 2^5 sign assignments.

Confidence intervals are reported as descriptive finite-sample intervals and are not used to conceal practically meaningful trade-offs that fail a conventional significance threshold.

6. Results

6.1 Main Policy Comparison

Table 4. Main results: mean ± std across five training seeds

Method	Cost	kg CO_2e	Fill %	Viol. %
Heuristic	17,846 ± 176	15,109 ± 72	95.15 ± 0.31	57.6 ± 7.9
Low-C heuristic	16,860 ± 156	13,826 ± 54	90.69 ± 0.31	4.4 ± 2.2
BC	17,727 ± 339	14,758 ± 97	93.85 ± 0.43	32.0 ± 5.1
Off. DQN	997,453 ± 85,925	2,353 ± 3,065	6.12 ± 1.97	0.0 ± 0.0
CQL	335,973 ± 114,523	9,574 ± 1,100	25.15 ± 9.47	0.0 ± 0.0
Support-CQL	16,639 ± 322	14,230 ± 72	92.80 ± 0.63	13.6 ± 6.2
CA-CQL	16,865 ± 404	14,119 ± 57	92.18 ± 0.88	0.0 ± 0.0

CA-CQL remains below the cap in all 250 main evaluation episodes and achieves 16,865 ± 404 cost, 14,119 ± 57 kg CO_2e, and 92.18 ± 0.88% fill. The standard heuristic provides the highest service, but exceeds the cap in 57.6% of episodes and has 5.49% higher cost and 6.55% higher emissions than CA-CQL. The low-carbon heuristic emits 292.9 kg less than CA-CQL and has essentially equal cost, but its fill rate is 1.48 percentage points lower and it still violates the cap in 4.4% of episodes. Thus, the hard cap is not equivalent to minimizing average emissions.

Relative to Support-CQL, CA-CQL reduces mean emissions by 110.8 kg (0.78%) and violation rate by 13.6 percentage points. The paired t 95% intervals are [−160.2, −61.4] kg and [−21.3, −5.9] percentage points, respectively. This compliance improvement costs 226.3 operating units (1.36%; interval [−47.5, 500.2]) and 0.626 fill-rate points (interval [−1.086, −0.167]). All five seed differences have the same sign; exact sign-flip p=0.0625 because of the small number of independent datasets. The evidence supports a stable direction of trade-off, but the cost effect should not be overstated.

Offline DQN and vanilla CQL fail operationally in this configuration. They select under-replenishing policies that reduce dispatch emissions but accumulate extreme backlog cost. Their inclusion is important: low emissions alone can be a symptom of service collapse. The result also illustrates why conservative value estimates without an explicit support-aware action selector can remain unsuitable in a large combinatorial discrete action space. It is not evidence that DQN or CQL always fail in inventory applications.

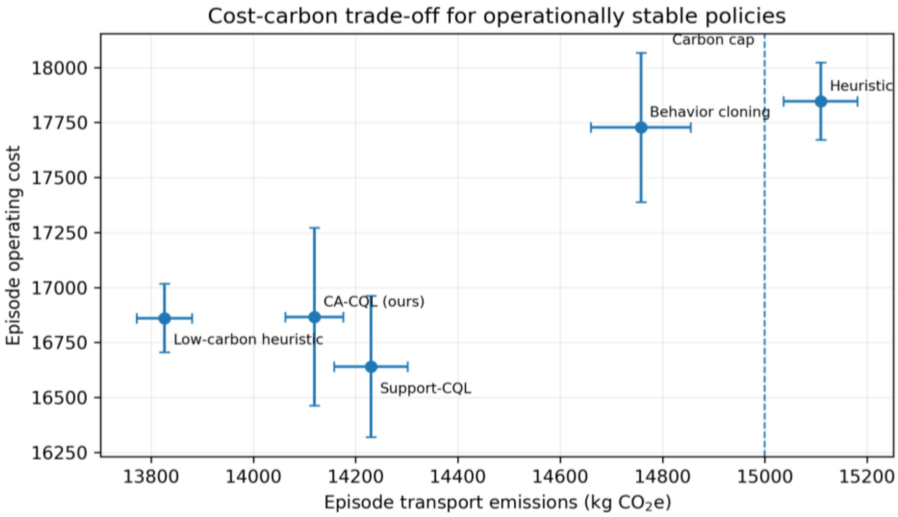

Figure 2. Cost–carbon trade-off among operationally stable policies; error bars are across five seed-level means

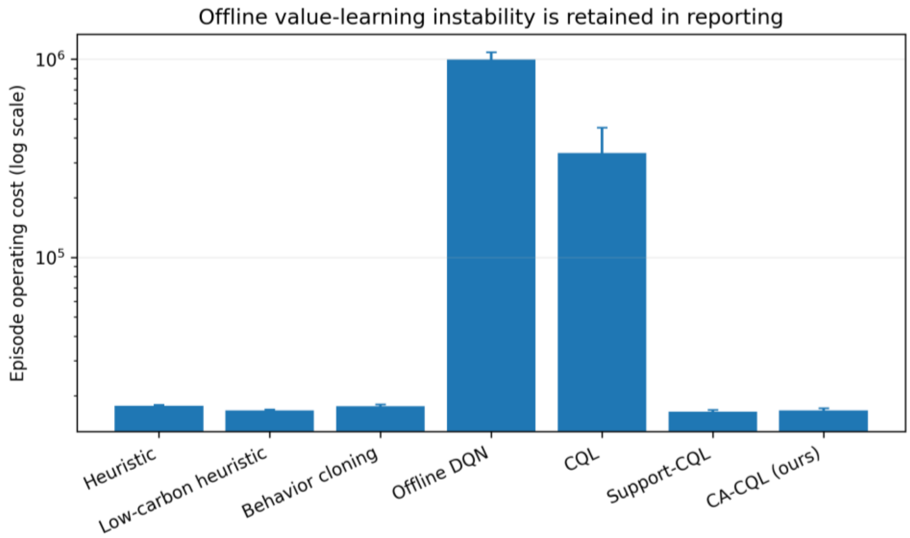

Figure 3. Total operating cost for all methods on a logarithmic axis. Offline DQN and vanilla CQL are retained rather than hidden as outliers

The logarithmic display makes the failure mode auditable: the weakly supported policies are not marginally worse but one to two orders of magnitude more costly. Reporting this figure together with fill rate and emissions prevents a misleading conclusion that a nearly inactive policy is environmentally superior.

6.2 Constraint Satisfaction

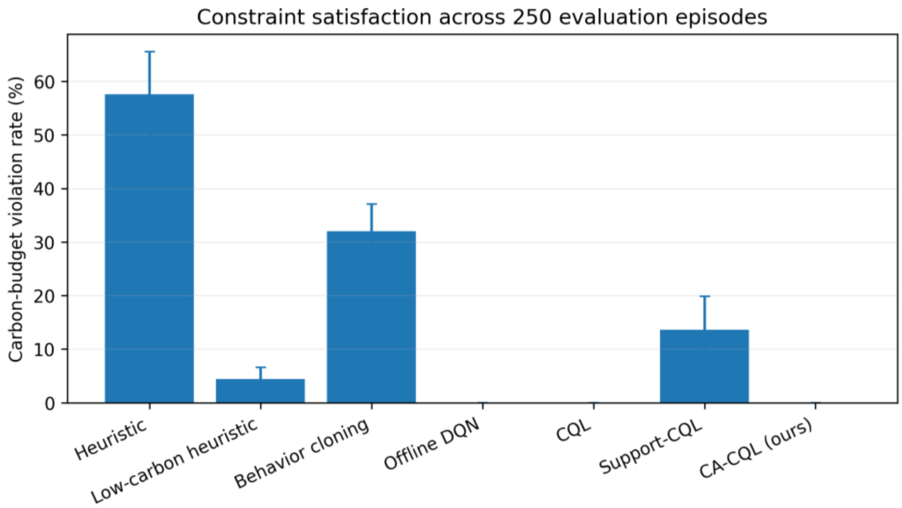

Figure 4. Carbon-budget violation rates over 250 episodes per method. Zero-emission collapse is not treated as good control

Figure 4 separates constraint satisfaction from average emissions. DQN and vanilla CQL have zero violation because they largely stop shipping, whereas CA-CQL combines zero violation with a 92.18% fill rate. The strict screen is therefore a compliance mechanism, while the value and behavior terms determine whether the cap-feasible policy remains operationally useful.

6.3 Component Ablation

Table 5. Deployment ablation results

Variant	Cost	kg CO_2e	Fill %	Viol. %
Full CA-CQL	16,568 ± 378	14,084 ± 139	92.66 ± 0.70	0.0 ± 0.0
Fixed C price	16,563 ± 377	14,083 ± 139	92.67 ± 0.69	0.0 ± 0.0
No C screen	16,581 ± 385	14,101 ± 146	92.66 ± 0.70	4.8 ± 3.0
No beh. prior	490,336 ± 132,793	7,942 ± 1,156	16.93 ± 5.51	0.0 ± 0.0

Removing the behavior prior causes a severe collapse: cost rises to 490,336 and fill falls to 16.93%. This confirms that the prior is not a cosmetic regularizer; it prevents the carbon-aware score from exploiting unsupported low-activity actions. Removing the carbon screen leaves average cost and service nearly unchanged but produces a 4.8% violation rate. Therefore, carbon shaping and pricing alone do not guarantee compliance. At the nominal cap, the fixed-price variant is nearly indistinguishable from the adaptive-price variant, so the experiment does not support a strong claim that adaptivity is essential in this scenario.

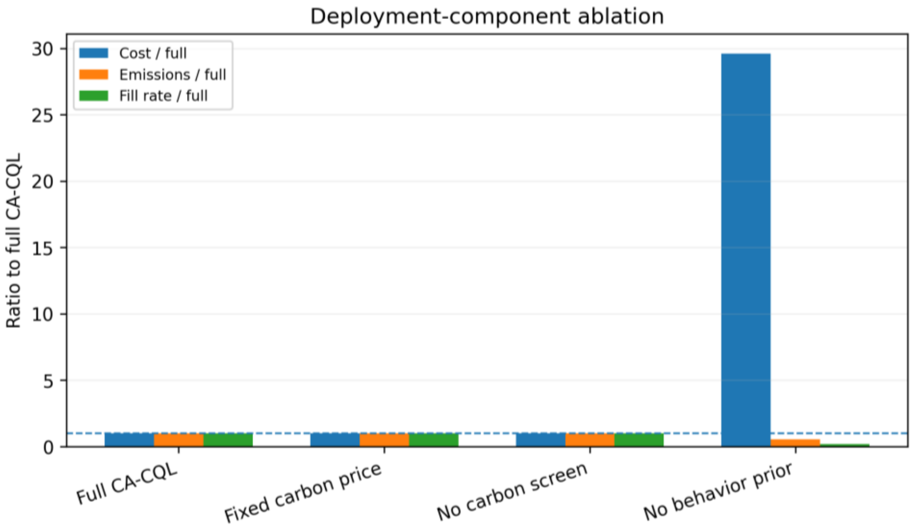

Figure 5. Ablation metrics normalized to full CA-CQL; the no-prior cost bar is clipped by the plotted scale and its exact value is reported in Table 5

The ablation also clarifies the novelty boundary. The hard screen alone can guarantee the modeled cap, but it cannot create a useful policy; the behavior prior alone can stabilize control, but it cannot enforce a cumulative carbon requirement. CA-CQL combines these two functions with a learned long-run value and a soft carbon allocation signal.

6.4 Zero-Shot Carbon-Cap Sensitivity

Table 6. Ca-cql sensitivity without retraining

Cap kg	Cost	kg CO_2e	Fill %	Viol. %
13,500	16,972 ± 508	13,410 ± 37	91.39 ± 0.79	0.0
15,000	16,590 ± 415	14,102 ± 99	92.69 ± 0.74	0.0
16,500	16,433 ± 427	14,173 ± 116	93.12 ± 0.67	0.0

The same trained policy respects all three caps. Tightening the cap from 15,000 to 13,500 kg lowers emissions to 13,410 kg, raises cost by about 382, and reduces fill from 92.69% to 91.39% in the sensitivity evaluation. Relaxing the cap to 16,500 kg improves fill to 93.12% and lowers cost, while emissions increase only modestly to 14,173 kg. The policy therefore does not automatically consume all available carbon; it uses the cap state as a constraint and economic signal rather than a target.

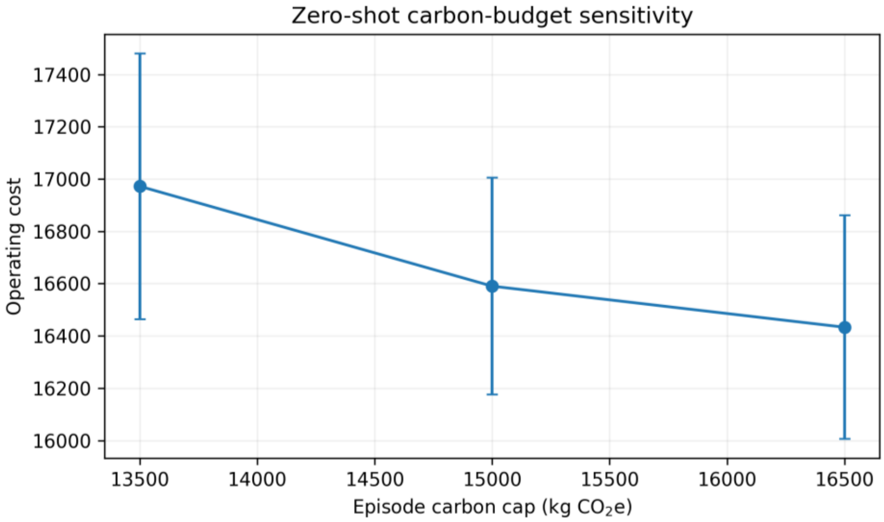

Figure 6. Operating-cost response to alternative carbon caps without retraining

7. Discussion and Limitations

7.1 Managerial Interpretation

The experiments suggest a practical separation of roles. The conservative Q-function estimates long-run operating value; the behavior prior discourages combinations absent from the log; the carbon price allocates scarce emissions over the horizon; and the screen enforces the nonnegotiable cap. This decomposition is interpretable for operations teams because each term corresponds to value, evidential support, carbon opportunity cost, or feasibility.

CA-CQL is most relevant where companies already possess fixed replenishment and dispatch logs but cannot safely explore new policies. Deployment requires mapping each candidate action to a credible pre-dispatch emission estimate. The method is compatible with Scope 1 fleet factors or Scope 3 freight factors when the accounting boundary is specified, but organizations should use factors and load data appropriate to their own geography, vehicle technology, and reporting standard [10-12].

7.2 Carbon Accounting Boundary and Deployment Checks

A carbon-aware controller is only as credible as its accounting boundary. Before deployment, an operator should define which legs, empty movements, subcontracted services, refrigeration loads, and well-to-tank components belong to the cap; map every discrete action to the corresponding distance, vehicle class, and load estimate; and retain the factor version used for each decision period. If the factor or distance is uncertain, a conservative upper estimate can be screened instead of the mean. This changes the compliance interpretation from "estimated emissions are below the cap" to "emissions under the selected accounting model and uncertainty margin are below the cap."

Chen et al. demonstrate that volatility-adaptive conformal prediction can attach calibrated uncertainty intervals to temporally dependent, nonstationary forecasts [33]. Their uncertainty-aware formulation suggests a valuable extension to CA-CQL: upper conformal bounds on demand or action emissions could replace point estimates in the feasibility screen when load and emission factors are uncertain.

The deterministic screen also creates a straightforward audit trail: the logged state contains remaining budget, the candidate list contains pre-dispatch emissions, and the selected action can be checked against the inequality $e(a) \le B - C_t$. Such traceability is useful even when a company ultimately replaces the present Q-network with another offline learner, because the guarantee depends on the screen and nonnegative accounting increments rather than on neural-network calibration.

7.3 Limitations and Threats to Validity

First, the operational environment is synthetic. It captures lead times, capacities, seasonality, overdispersion, backlog, consolidation, express service, and load-sensitive emissions, but it is not calibrated to a named firm. External validity requires testing on an enterprise log that includes decisions and transportation activity, not sales alone. Second, the network has one product and three retailers; larger SKU and routing spaces may require factorized, hierarchical, or multi-agent policies.

Third, the carbon model covers modeled road-freight vehicle-kilometer emissions. It excludes warehouse energy, manufacturing, vehicle production, refrigeration, empty repositioning outside the specified routes, and other lifecycle sources. Fourth, linear interpolation between official 0%- and 100%-laden factors is a transparent approximation, not a universal physical law. Fifth, hyperparameters were selected for a compact reproducible run rather than through a large validation campaign. Sixth, five independent training datasets are sufficient to expose consistent directional effects but provide coarse nonparametric inference. Finally, the feasibility screen can preserve a cap by choosing no shipment, so service metrics must always be reported alongside compliance.

7.4 Reproducibility and Data Availability

The deliverables include Python source, exact command lines, five compressed offline datasets, seed-level and summary CSV files, statistical tests, ablations, sensitivity results, a trained seed-0 checkpoint, figures, the official factor workbook and extracted rows, reference verification, and SHA-256 hashes. Re-running can produce small numerical differences across software versions; the provided CSV files are the outputs used in this manuscript. No proprietary or personally identifiable data are included.

8. Conclusion

This paper introduced CA-CQL for joint replenishment and transportation scheduling under a finite carbon budget using only fixed offline data. The method augments conservative Q-learning with budget state, carbon-shaped rewards, a behavior-support prior, a state-dependent carbon price, and a deterministic action screen. In the reproducible multi-echelon benchmark, CA-CQL eliminated carbon-cap violations while retaining useful service and competitive cost. Relative to Support-CQL, the gain in compliance and emissions came with a measurable service–cost trade-off. Ablations identified the behavior prior and carbon screen as the decisive components, while the adaptive price had limited effect at the nominal cap. Future work should evaluate logged enterprise decisions, larger multi-product networks, richer routing, uncertainty in emission factors, and offline policy selection that does not require simulator evaluation.

References

[1] Clark, A. J., & Scarf, H. (1960). Optimal policies for a multi-echelon inventory problem. Management Science, 6(4), 475–490. https://doi.org/10.1287/mnsc.6.4.475

[2] Campbell, A. M., Clarke, L., Kleywegt, A. J., & Savelsbergh, M. W. P. (1998). The inventory routing problem. In T. G. Crainic & G. Laporte (Eds.), Fleet management and logistics (pp. 95–112). Springer. https://doi.org/10.1007/978-1-4615-5755-5_4

[3] Trudeau, P., & Dror, M. (1992). Stochastic inventory routing: Route design with stockouts and route failures. Transportation Science, 26(3), 171–184. https://doi.org/10.1287/trsc.26.3.171

[4] Kleywegt, A. J., Nori, V. S., & Savelsbergh, M. W. P. (2002). The stochastic inventory routing problem with direct deliveries. Transportation Science, 36(1), 94–118. https://doi.org/10.1287/trsc.36.1.94

[5] Kleywegt, A. J., Nori, V. S., & Savelsbergh, M. W. P. (2004). Dynamic programming approximations for a stochastic inventory routing problem. Transportation Science, 38(1), 42–70. https://doi.org/10.1287/trsc.1030.0041

[6] Campbell, A. M., & Savelsbergh, M. W. P. (2004). A decomposition approach for the inventory-routing problem. Transportation Science, 38(4), 488–502. https://doi.org/10.1287/trsc.1030.0054

[7] Cheng, C., Qi, M., Wang, X., & Zhang, Y. (2016). Multi-period inventory routing problem under carbon emission regulations. International Journal of Production Economics, 182, 263–275. https://doi.org/10.1016/j.ijpe.2016.09.001

[8] Bektaş, T., & Laporte, G. (2011). The pollution-routing problem. Transportation Research Part B: Methodological, 45(8), 1232–1250. https://doi.org/10.1016/j.trb.2011.02.004

[9] Demir, E., Bektaş, T., & Laporte, G. (2014). A review of recent research on green road freight transportation. European Journal of Operational Research, 237(3), 775–793. https://doi.org/10.1016/j.ejor.2013.12.033

[10] World Resources Institute, & World Business Council for Sustainable Development. (2011). Corporate value chain (Scope 3) accounting and reporting standard.

[11] Smart Freight Centre. (2023). Global logistics emissions council framework for logistics emissions accounting and reporting (Version 3.0).

[12] UK Department for Energy Security and Net Zero, & Department for Environment, Food & Rural Affairs. (2026). Greenhouse gas reporting: Conversion factors 2026.

[13] Altman, E. (1999). Constrained Markov decision processes. CRC Press.

[14] Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.

[15] Achiam, J., Held, D., Tamar, A., & Abbeel, P. (2017). Constrained policy optimization. In Proceedings of the 34th International Conference on Machine Learning (pp. 22–31).

[16] Fujimoto, S., Meger, D., & Precup, D. (2019). Off-policy deep reinforcement learning without exploration. In Proceedings of the 36th International Conference on Machine Learning (pp. 2052–2062).

[17] Kumar, A., Fu, J., Soh, M., Tucker, G., & Levine, S. (2019). Stabilizing off-policy Q-learning via bootstrapping error reduction. In Advances in Neural Information Processing Systems 32.

[18] Kumar, A., Zhou, A., Tucker, G., & Levine, S. (2020). Conservative Q-learning for offline reinforcement learning. In Advances in Neural Information Processing Systems 33 (pp. 1179–1191).

[19] Fu, J., Kumar, A., Nachum, O., Tucker, G., & Levine, S. (2021). D4RL: Datasets for deep data-driven reinforcement learning. In Advances in Neural Information Processing Systems 34 (pp. 4933–4946).