Latency-Aware Bid Acceptance under Operational Feasibility: A Public Benchmark with Hindsight Ceilings*

Aswin Chandrasekaran
Bubba AI
aswin@bubba.ai

July 2026

Abstract

Online truckload bid acceptance is a closed-loop stochastic decision problem in which a carrier or broker must, in real time, accept or reject a tendered load subject to operational feasibility, fleet repositioning costs, and opportunity cost against future demand. Public, reproducible benchmarks for this problem are scarce: existing routing benchmarks are static, while dynamic-fleet studies typically rely on private operator data. We introduce FreightBidBench, a public-calibrated, dependency-free, closed-loop benchmark in which feasibility (pickup reach, appointment windows, simplified hours-of-service, stochastic yard delays) and economics (service-failure penalty, terminal fleet value, daily price-premium window) are explicit, versioned, and reproducible from public Freight Analysis Framework and U.S. Department of Agriculture truck-rate data. Building on this benchmark, we make three contributions. First, we formalize the v0.3 closed-loop accept/reject MDP and show that the three new reward components each isolate a distinct policy class: a service-failure penalty creates linear regret for feasibility-blind policies, terminal fleet value penalizes greedy positioning, and a temporal price-premium window penalizes future-blind timing. Second, we develop three complementary hindsight diagnostics: an exact realized-seed dynamic program for small load prefixes, a simple LP-style full-horizon upper bound, and a Lagrangian-per-truck information-relaxation bound that retains per-truck HOS and sequencing structure and is 20.7% tighter than the LP relaxation on `tight` and 39.3% tighter on `scarce` while remaining dependency-free, justified as an information relaxation in the sense of Brown et al. [2010]. Third, we introduce a parametric surrogate-rollout cascade with two escalation triggers — a boundary band $\beta \geq 0$ on the surrogate's signed score and a scarcity-pressure threshold κ on the count of immediately available trucks in the origin market — characterize its limit behaviour (rollout-call share monotone in either trigger threshold), and show that the scarcity trigger alone captures the high-stakes capacity decisions on which the surrogate is least reliable. On ten-seed tight and scarce scenarios, the best simple policy retains 91.0 percent and 86.5 percent of rollout profit, respectively, and a stdlib linear surrogate 94.2 percent and 89.3 percent; a cascade at a single escalation band recovers roughly 98 percent of rollout value on both at 40–56 percent of rollout's mean decision latency, and on `tight` is statistically indistinguishable from the rollout teacher (paired-bootstrap 95% CI on the profit delta spans zero). The release contains a versioned manifest, layer ablations, sensitivity sweeps, and an exact-plus- relaxed ceiling, providing a reproducible test bed for future methods work.

*Code and benchmark artifacts: `https://github.com/aswincsekar/freightbidbench`. Artifact release: `freightbidbench-v0.3`, scenario contract `scenario-v0.3.2`, policy set `policy-set-v0.3.0`.

1 Introduction

Truckload carriers and brokers face an online accept/reject decision each time a load tender arrives. The decision is constrained by latency: tenders in a typical broker workflow must be priced and accepted or rejected within seconds. The decision is operationally constrained: a load can only be served by a truck that can reach the pickup, satisfies appointment windows and hours-of-service (HOS) clocks, and remains feasible through delivery. The decision is economically constrained: a tender with positive immediate margin can still be a poor decision if it consumes a scarce truck before a predictable price-premium window or strands the fleet in a market with weak outbound flow. Conversely, no benchmark can claim to compare future-aware acceptance policies meaningfully unless it (a) makes operational feasibility part of the reward function rather than a side diagnostic, and (b) anchors policy evaluation against a measured ceiling rather than only against a finite stochastic rollout teacher.

Existing routing benchmarks, such as the Solomon and Homberger–Gehring VRPTW instances [Solomon, 1987, Homberger and Gehring, 2005] and the CVRPLIB tradition [Uchoa et al., 2017], anchor the static vehicle-routing literature but are not closed-loop and do not model online stochastic tender arrivals. Stochastic and dynamic vehicle-routing surveys [Pillac et al., 2013, Ritzinger et al., 2016] document the distinction between offline policy construction and online computation, while fleet-management approximate dynamic programming (ADP) work [Simão et al., 2009, Powell et al., 2012, 2014, Topaloglu and Powell, 2006] has historically been driven by private operator data. Recent stochastic VRP benchmarks [Heakl et al., 2025] target a different problem geometry (multi-customer routing) than online truckload bid acceptance. The closed-loop, fleet-state-dependent, latency-aware truckload accept/ reject problem with public calibration has lacked a shared reproducible artifact.

FreightBidBench v0.2 took a first step by adding operational feasibility (per-truck records, pickup reach time, appointment windows, simplified 11/14/10 HOS, stochastic yard delays) to a public-calibrated benchmark built on the Freight Analysis Framework [U.S. Department of Transportation, Bureau of Transportation Statistics and Federal Highway Administration, 2017] and U.S. Department of Agriculture truck-rate reports [U.S. Department of Agriculture, Agricultural Marketing Service, 2026]. That release showed that feasibility is first-order: ignoring HOS or appointment windows materially changes which policies look competitive. However, the v0.2 economic structure was too flat for a credible methods claim: simple greedy baselines and the finite rollout teacher were often within a few percentage points of each other, leaving no room for cheaper future-aware approximations to demonstrate value.

The present paper, anchored on the v0.3 release of FreightBidBench, addresses three gaps. We sharpen the benchmark economics so that the decision problem has provable structural separation between feasibility- blind, future-blind, and future-aware policies; we anchor policy comparison against both an exact small-prefix dynamic program and a dependency-free relaxed full-horizon upper bound; and we introduce a parametric latency- aware policy class that interpolates between a cheap surrogate and a finite rollout teacher. Throughout, the benchmark is the primary artifact: the methods results are evidence that the sharpened problem admits a non-trivial latency-profit frontier, but the benchmark stands as a reproducible test bed independent of any specific method.

Our contributions are as follows. We formalize the v0.3 closed-loop MDP and show that each of three new reward components (service-failure penalty, terminal fleet value, temporal price-premium window) isolates a distinct class of policies that any future-aware method must beat; for the service-failure penalty this separation is an exact identity (Proposition 1), and for terminal value and the price-premium window it is established empirically through layer ablations. We characterize two complementary full-horizon upper bounds: a simple LP-style relaxation that drops integrality, sequencing, and location constraints, and a tighter Lagrangian-per-truck information

relaxation [Brown et al., 2010] that dualizes only the cross-truck assignment constraint and retains per-truck HOS, location, and sequencing structure. We decompose the LP looseness into positioning, sequencing, and integrality components and show that the Lagrangian bound recovers most of the sequencing and integrality slack. We define a surrogate-rollout cascade as a parametric policy class with boundary-band and scarcity-pressure escalation triggers, prove its limit behaviour, and provide a structural justification for the scenario-dependent calibration that we observe empirically. Finally, we release the benchmark with versioned scenario, policy, and feasibility contracts; a manifest-based reproducibility protocol; layer-ablation and sensitivity experiments; and exact-plus-relaxed hindsight ceilings.

2 Related Work

Dynamic truckload fleet management and bid acceptance. Dynamic load acceptance and rejection in truckload operations has been formulated as an online state-dependent control problem since at least Kim et al. [2004], who study large-fleet acceptance with priority demand and time windows. Yang et al. [2004] study real-time multi-vehicle truckload pickup and delivery and develop online reoptimization heuristics. Tjokroamidjojo et al. [2006] quantify the value of advance load information in truckload trucking, characterizing how lookahead and information structure change profitability. The ADP-based fleet-management line, beginning with Simão et al. [2009] and surveyed in Powell et al. [2012], established that state-dependent value-function approximation can match operator-scale fleet decisions. Subsequent applications include locomotive optimization [Powell et al., 2014] and time-staged integer multi-commodity flow [Topaloglu and Powell, 2006]. FreightBidBench complements this line by providing a reproducible benchmark on which policies of varying complexity — including ADP-style value-function approximations, finite rollout teachers, and lightweight surrogates — can be compared under shared stochastic seeds.

Stochastic and dynamic vehicle routing. Surveys by Pillac et al. [2013] and Ritzinger et al. [2016] catalogue dynamic and stochastic vehicle-routing problems and the distinction between offline policy construction and online computation. Ulmer et al. [2019] and Ulmer and Thomas [2020] develop offline–online ADP and meso-parametric value-function approximations for dynamic customer acceptance. Secomandi and Margot [2009] study reoptimization for the VRP with stochastic demands. These works motivate the rollout teacher and surrogate tracks in FreightBidBench. The benchmark gap is that truckload bid evaluation also requires public freight calibration, fleet repositioning under operational feasibility, and latency-profit reporting — none of which are first-class in static VRPTW instance sets.

Hindsight bounds and information relaxation. Anchoring policy evaluation against an upper bound is a long-standing theme in stochastic dynamic programming. Brown et al. [2010] develop a general information-relaxation duality framework that gives upper bounds on stochastic-DP optima by giving the decision maker access to future information offset by an optional penalty. Adelman and Mersereau [2008] develop Lagrangian relaxations for weakly coupled stochastic dynamic programs. Both lines justify reporting a ceiling alongside achievable policies. We adopt the information-relaxation framing for the v0.3 relaxed full-horizon bound and report it explicitly as a (loose) upper bound rather than a feasible plan. The exact realized-seed dynamic program we report on small load prefixes serves as the trustworthy reference against which the relaxation can be calibrated.

Rollout policies and cascades. Rollout, introduced for combinatorial optimization by Bertsekas et al. [1997] and extended for finite-horizon stochastic DPs by Goodson et al. [2017], is the methodological lineage of the v0.3 rollout teacher: common-random-number Monte Carlo expansion of accept and reject branches, with the better expected branch chosen. Rollout is accurate but computationally expensive. Cascading a cheap surrogate to an expensive teacher, escalating only when the surrogate is uncertain, instantiates two older ideas: cascade classifiers, which route easy instances through cheap stages and hard instances to expensive stages [Viola and Jones, 2001], and anytime algorithms and metareasoning, which allocate computation according to its expected value [Boddy and Dean, 1989, Zilberstein, 1996, Russell and Wefald, 1991, Horvitz, 1988]. What is missing for stochastic decision problems is a systematic benchmark of such cascades under public calibration with operational feasibility. The v0.3 paper formalizes the cascade as a parametric policy class indexed by escalation triggers and reports its frontier on the public benchmark.

Learning-augmented routing. Recent learning-augmented optimization studies use neural or attention- based policies to accelerate routing decisions [Nazari et al., 2018, Kool et al., 2019, Patel et al., 2022]. We intentionally keep the v0.3 reference implementation dependency-free so that the benchmark itself remains reproducible from the Python standard library; optional learned baselines are deferred to a stretch track in future releases.

Public benchmarks in OR. Static benchmark instances have shaped routing research for decades [Solomon, 1987, Homberger and Gehring, 2005, Uchoa et al., 2017]. Recent stochastic routing benchmarks [Heakl et al., 2025] push toward dynamic settings but target multi-customer pickup-and-delivery rather than online truckload bid acceptance. FreightBidBench fills the latter gap with versioned scenario, policy, and feasibility contracts and a manifest-based reproducibility protocol.

3 Problem Formulation

We formalize closed-loop truckload bid acceptance as a finite-horizon Markov decision process (MDP) with continuous time-stamped events.

3.1 State and Action

Let $\mathcal{T} = [0, T]$ denote the horizon (in hours; $T = 72$ for the v0.3 scenarios). The fleet consists of K trucks, each with state

$$u_t^{(k)} = \big(\ell_t^{(k)},\ \tau_t^{(k)},\ h_t^{(k)},\ d_t^{(k)}\big), \tag{1}$$

where $\ell_t^{(k)} \in \mathcal{S}$ is the truck's market (state), $\tau_t^{(k)}$ is the next available time, $h_t^{(k)}$ is the remaining HOS drive-time budget, and $d_t^{(k)}$ is the remaining HOS duty-time budget. The fleet state at time t is $F_t = (u_t^{(1)}, \ldots, u_t^{(K)})$.

A load tender event at time t presents a candidate load ℓ_t characterized by an attribute tuple

$$\ell_t = \big(o, d, p, c_{\text{lin}}, m, \tau_{\text{e}}^p, \tau_{\text{l}}^p, \tau_{\text{e}}^d, \tau_{\text{l}}^d, Y^p, Y^d\big), \tag{2}$$

where $o, d \in \mathcal{S}$ are origin and destination markets, p is the posted price (after the temporal premium described below), c_{lin} is the linehaul cost, m is the linehaul mileage, $\tau_{\text{e}}^p, \tau_{\text{l}}^p$ are pickup appointment window endpoints, $\tau_{\text{e}}^d, \tau_{\text{l}}^d$ are delivery appointment window endpoints, and Y^p, Y^d are stochastic yard-delay draws at pickup and dropoff. The system state at decision time t is $s_t = (F_t, \ell_t, t)$ and the action space is binary, $a_t \in \mathcal{A} = \{0, 1\}$ (reject or accept).

3.2 Feasibility Layer

A deterministic feasibility map $\Psi : (F, \ell) \to (k^\star, F', \text{flag})$ attempts to assign load ℓ to a truck. The candidate set is the trucks in the origin market o; candidates whose next-available time already exceeds the load's latest pickup are dropped. For each remaining candidate the map computes pickup-reach time, pickup-arrival time including yard delay Y^p, drive time over m miles, HOS rest insertions if required, and delivery-arrival time with yard delay Y^d. It returns $k^\star = \emptyset, F' = F, \text{flag} = \texttt{infeasible}$ if no truck admits an HOS-feasible schedule within τ_{l}^p and τ_{l}^d. Otherwise, among the feasible candidates it selects the truck that becomes available earliest after delivery (minimizing the post-delivery next-available time, ties broken by fleet index), and returns that truck, the updated fleet state, and flag = `ok`.

3.3 Reward

For action $a_t = 1$ and feasibility flag `ok`:

$$r_t(s_t, 1) = p - c_{\text{lin}} - c_{\text{ph}} \cdot m_{\text{dh}}^p - c_Y \cdot (Y^p + Y^d), \tag{3}$$

where c_{ph} is the cost per deadhead mile, m_{dh}^p is the pickup- reach mileage from the assigned truck, and c_Y is the per-hour yard-delay cost. For action $a_t = 1$ and feasibility flag `infeasible`, the fleet is not mutated and the reward is the service-failure penalty:

$$r_t(s_t, 1) = -\rho, \tag{4}$$

where $\rho \geq 0$ is the v0.3 service-failure penalty (Section 4). For $a_t = 0$, $r_t = 0$.

At the horizon the system collects a terminal fleet reward

$$\Phi(F_T) = \omega \sum_{k=1}^{K} V(\ell_T^{(k)}), \tag{5}$$

where $\omega \geq 0$ is the v0.3 terminal value weight and $V(\cdot)$ is a state-value signal computed once per scenario from public data: with $b(\ell)$ the FAF outbound tonnage of state ℓ and $g(\ell)$ its net outbound-tonnage imbalance,

$$V(\ell) = \sigma\left(0.70 \cdot \hat{b}(\ell) + 0.30 \cdot \hat{g}(\ell)\right), \quad \hat{b}(\ell) = \frac{2b(\ell)}{\max_j b(j)} - 1, \quad \hat{g}(\ell) = \frac{g(\ell)}{\max_j |g(j)|}, \tag{6}$$

where σ is the scenario terminal value scale (`value_scale_dollars`). The 0.70/0.30 split weights raw outbound demand above the directional imbalance correction.

3.4 Objective

A policy π is a mapping from system state s_t to a probability over actions. The closed-loop objective is to maximize expected total reward

$$V^\pi = \mathbb{E}\left[\sum_{t:\ell_t \text{ tendered}} r_t(s_t, \pi(s_t)) + \Phi(F_T)\right], \tag{7}$$

under the joint distribution of load arrivals (Poisson with hour-of-day modulated rate $\lambda(t)$), candidate-load draws from the public-calibrated lane table, and yard-delay distributions. We compare policies under common random numbers: for a fixed (train seed, eval seed) pair, load streams, yard delays, and initial fleet positions are identical across all policies.

3.5 Notation Summary

Throughout, $V^\star = \sup_\pi V^\pi$ denotes the optimal expected reward; V^R denotes the finite-lookahead rollout teacher's expected reward; V^S_θ denotes a surrogate policy parameterized by θ; and V^β_θ denotes the cascade with escalation band β built from surrogate θ and rollout teacher. A realized scenario (one common-random-number sample path) is written ξ; the load tuple is ℓ_t and truck k's market is $\ell_t^{(k)}$. The reward parameters are the service-failure penalty ρ, the terminal value weight ω, and the terminal value scale σ; the cascade is indexed by the boundary band β and scarcity threshold κ, with $n_o(s)$ the count of immediately available origin-market trucks and $\Delta_\theta(s)$ the surrogate's signed accept-minus-reject score.

4 Benchmark Economics in v0.3

The v0.3 release introduces three reward components on top of the v0.2 feasibility-aware MDP. Each component is independently controllable through the versioned scenario contract `configs/freightbidbench_v03_scenarios.json`, currently frozen at `scenario-v0.3.2`, and each isolates a distinct policy class.

4.1 L1: Service-Failure Penalty

A policy that returns $a_t = 1$ on a load that the feasibility map classifies as infeasible incurs the reward $-\rho$ with $\rho \geq 0$. Fleet state is not mutated. In v0.2, $\rho = 0$ and failed accepts were recorded only as diagnostic events; in v0.3, $\rho = \$10$ after calibration.

The following observation justifies ρ's role.

Proposition 1 (Feasibility-blind regret under L1)**.** *Let π be a policy that does not consult the feasibility map before returning $a_t = 1$, and let π' be its feasibility-aware variant that returns $a_t = 1$ iff π returns $a_t = 1$* and *the feasibility flag is `ok`. Let $N(\pi, \xi)$ denote the realized number of infeasible accepts of π on scenario ξ. Then*

$$V^{\pi'} - V^\pi = \rho \cdot \mathbb{E}_\xi[N(\pi, \xi)]. \tag{8}$$

Proof. By construction, π and π' agree on every decision except infeasible attempts. Fleet state is not mutated on infeasible accepts, so π' and π have identical fleet trajectories and identical realized rewards except for the infeasible-accept events, at which π pays $-\rho$ per event while π' pays 0. □

Proposition 1 establishes that $\rho > 0$ creates linear regret in the realized infeasible-accept count. Empirically, $\rho = \$10$ is the smallest tested value that flips the per-policy ordering of `myopic_margin` and `bid_price` below `accept_all_feasible` on both `tight` and `scarce` (Section 8).

4.2 L2: Terminal Fleet Value

End-of-horizon trucks receive value $\omega \cdot V(\ell_T^{(k)})$ as in Equation (5), with $\omega = 0.25$ in `scenario-v0.3.2`. This term penalizes policies that strand trucks in low-value markets at horizon end. Unlike L1, L2 acts on policies that *respect feasibility* but are myopic with respect to terminal positioning. Specifically, the feasibility-aware greedy baseline `accept_all_feasible` loses retention against rollout under L2 because it accepts the first feasible load without consideration of where the truck ends.

4.3 L3: Temporal Price-Premium Window

The frozen demand-wave schedule keeps the load-arrival rate flat and applies a daily multiplicative price premium $\mu(t) > 1$ during hours $t \bmod 24 \in [8, 16)$ and $\mu(t) = 1$ otherwise. In `scenario-v0.3.2`, $\mu = 1.5$ in-window. The schedule creates a controlled timing problem: accepting a moderate off-peak load can consume capacity required for a predictable on-peak premium. L3 separates *future-blind on timing* policies, including `accept_all_feasible` and the static `bid_price` baseline (which uses an aggregated origin–destination future-value proxy without hour-of-day awareness), from future-aware policies that can reason about premium-window saturation.

4.4 Versioned Artifacts

The v0.2 reference run remains immutable under `benchmark_runs/paper_v02/`; the v0.3 scenario contract lives in `configs/freightbidbench_v03_scenarios.json`; and v0.3 table drafts and ceilings are assembled under `benchmark_runs/paper_v03/`. The versioning protocol (Section A) requires that any change to scenario parameters, the default policy set, or the cascade band schedule bump the relevant version string in the manifest.

5 Hindsight Ceilings

We anchor policy comparison against three complementary ceilings: an exact realized-seed dynamic program on small load prefixes, a simple LP-style relaxed full-horizon upper bound, and a Lagrangian-per-truck information-relaxation bound (Proposition 3) that respects per-truck sequencing and HOS structure. The exact DP is the trustworthy small-instance reference; the LP relaxation is a fast but loose full-horizon ceiling; the Lagrangian-per-truck bound is substantially tighter while remaining dependency-free.

5.1 Exact Small-Prefix Dynamic Program

For a truncated realized stream of L loads, the exact problem is a binary decision tree of depth L over the deterministic post-acceptance fleet transition. We solve it by memoized search keyed on $(\text{prefix index}, \text{fleet hash})$. The state count grows exponentially in L but the deterministic assignment rule and small fleet keep the search tractable for $L \leq 16$ on the v0.3 scenarios. The implementation is in `scripts/run_hindsight_bound.py`.

The exact DP is the trustworthy small-instance reference; we report it together with the relaxed bound rather than as a substitute.

5.2 Relaxed Full-Horizon Upper Bound

For the full horizon, we report the minimum of two upper bounds plus a terminal upper bound:

1. **Positive-profit relaxation.** Accept every realized load with positive fresh-truck profit, ignoring fleet location, sequencing, and capacity.

2. **Fractional truck-hour relaxation.** Ignore location and sequencing, charge each profitable load a lower-bound busy time, and solve the resulting fractional truck-hour knapsack.

Both bounds additively include a terminal upper bound that allows every truck to end in the highest-value market.

Proposition 2 (Validity of the relaxed bound)**.** *Let $U^R(\xi)$ denote the relaxed full-horizon objective evaluated on realized scenario ξ. Then*

$$\mathbb{E}_\xi\big[\, U^R(\xi)\,\big] \;\geq\; V^\star. \tag{9}$$

Sketch. U^R is constructed by (i) revealing the full realized scenario ξ in advance (a zero-penalty information relaxation in the sense of Brown et al. [2010]), (ii) replacing the integer truck-assignment constraint by its fractional relaxation, (iii) dropping the location and sequencing constraints, and (iv) replacing the terminal fleet position by the best-case state value. The information relaxation yields an upper bound: $\mathbb{E}[\sup_{a(\omega)} V(a,\omega)] \geq \sup_\pi \mathbb{E}[V(\pi(\omega),\omega)]$ [Brown et al., 2010]. Each of (ii)–(iv) further enlarges the feasible set, hence the resulting expectation only increases. □

Remark 1 (Decomposition of looseness)**.** *The gap $U^R - V^\star$ decomposes into four components: information gain (allowing the decision maker to see future loads), integrality loss (fractional truck-hours), sequencing loss (dropping the constraint that a truck can only carry one load at a time in a feasible spatial sequence), and terminal optimism (best-case terminal positioning). In the v0.3 scenarios the sequencing and integrality components dominate; the terminal upper bound contributes a small fraction of total looseness.*

The bound is reported with explicit caveats. We do not interpret the relaxed-bound retention numbers as achievable performance; they characterize the structural hardness of the problem and bound the methodological headroom for future work.

5.3 Lagrangian-per-truck Information Relaxation

The LP-style relaxation of Section 5.2 throws away three constraints simultaneously: per-truck location continuity, per-truck sequencing, and integrality. Most of its looseness comes from the location/sequencing relaxation, since real truck schedules are spatially constrained. We tighten the bound by retaining per-truck structure and dualizing only the cross-truck assignment constraint.

Recall from Section 3 that the joint problem can be written as

$$V^\star = \sup_a \mathbb{E}\left[\sum_t r_t(s_t,a_t) + \Phi(F_T)\right] \quad \text{s.t.} \quad \sum_{k=1}^{K} a_t^{(k)} \leq 1 \text{ for each tendered load } t, \tag{10}$$

where $a_t^{(k)} \in \{0,1\}$ indicates whether truck k is assigned to load t under the joint action a_t and the per-load reward decomposes as $r_t = \sum_k r_t^{(k)} a_t^{(k)}$. Introduce non-negative duals $\boldsymbol{\lambda} = \{\lambda_t \geq 0\}$ on the assignment constraints and form the Lagrangian

$$L(\boldsymbol{\lambda}) = \sum_t \lambda_t + \sum_{k=1}^{K} V_{\boldsymbol{\lambda}}^k\big(u_0^{(k)}\big), \tag{11}$$

where the per-truck sub-MDP value is

$$V_{\boldsymbol{\lambda}}^k(u_0^{(k)}) = \sup_{a^{(k)}} \mathbb{E}\left[\sum_t a_t^{(k)}\big(r_t^{(k)} - \lambda_t\big) + \omega\, V(\ell_T^{(k)})\right], \tag{12}$$

with the supremum taken over per-truck policies that respect the truck's own HOS clocks, pickup-reach time, appointment windows, and location continuity.

Proposition 3 (Lagrangian-per-truck upper bound)**.** *For any* $\boldsymbol{\lambda} \in \mathbb{R}^{|T|}_{\geq 0}$, $V^\star \leq L(\boldsymbol{\lambda})$. *Consequently*

$$V^\star \leq \inf_{\boldsymbol{\lambda} \geq 0} L(\boldsymbol{\lambda}), \tag{13}$$

and the infimum is attained at some $\boldsymbol{\lambda}^\star \geq 0$*: on a realized scenario each per-truck sub-MDP has finitely many policies, so* L *is a finite maximum of affine functions of* $\boldsymbol{\lambda}$ *(piecewise linear and convex) and coercive on* $\boldsymbol{\lambda} \geq 0$ *(as* $\lambda_t \to \infty$ *every truck rejects load* t*, so* $L \to \infty$*).*

Sketch. The Lagrangian dualizes the cross-truck assignment constraint $\sum_k a_t^{(k)} \leq 1$ without dropping any per-truck constraint. For any $\boldsymbol{\lambda} \geq 0$, the supremum over assignments without the constraint upper-bounds the supremum over feasible assignments (weak duality). Decomposability of r_t across trucks splits the relaxed supremum into a sum of independent per-truck suprema (Equation 11). Convexity of L in $\boldsymbol{\lambda}$ follows from the pointwise-supremum-of-affine characterization. □

Compared to Proposition 2's LP-style bound, the Lagrangian relaxation drops only one constraint (cross-truck assignment) rather than three, so it is tighter. We minimize L over $\boldsymbol{\lambda} \geq 0$ by projected subgradient descent: a subgradient component on λ_t is $1 - \sum_k a_t^{(k)\star}(\boldsymbol{\lambda})$, where $a_t^{(k)\star}(\boldsymbol{\lambda})$ is truck k's optimal acceptance of load t in its sub-MDP, so the diminishing-step update that descends L and projects onto $\boldsymbol{\lambda} \geq 0$ is $\lambda_t \leftarrow \max(0, \lambda_t + \alpha_n(\sum_k a_t^{(k)\star} - 1))$ with $\alpha_n = c/\sqrt{n}$.

Computational construction. Each per-truck sub-MDP is solved exactly by forward enumeration of reachable states under common random numbers (load arrivals and yard delays are deterministic on the realized scenario). The continuous state space (market, available-time, drive-used, duty-used) is quantized into buckets at 15-minute time and 1-hour HOS resolutions; on entry to a bucket, clocks are snapped to the bucket's favorable lower corner. The snapping is permissive (more time and HOS budget available), so per-truck DP values can only over-estimate the exact per-truck Lagrangian sup, preserving the joint upper-bound guarantee. Within-bucket dominance reduces to value comparison; cross-bucket Pareto pruning operates on bucket indices.

Convergence and empirical bound. On the `tight` scenario at eval seed `20260507` (995 realized loads, fleet size 70), 30 cold-start subgradient iterations at step scale 100 followed by 20 warm-start iterations from the iter-30 duals produced the best bound $L^\star = \$1{,}885{,}043$. The bound trajectory was monotone decreasing through iter ∼40 and oscillated within ±$10k thereafter. Compared with the LP-style bound of $2,377,500 on the same seed, the Lagrangian bound is 20.7% tighter. It remains a valid upper bound: the rollout teacher's realized profit on the same seed is $1,273,395 $< L^\star$. On the `scarce` scenario at the same eval seed (1,154 realized loads, fleet size 55), 30 cold-start iterations at step scale 100 produced $L^\star = \$1{,}623{,}084$, 39.3% tighter than the LP-style bound of $2,675,525 and still valid against rollout ($1,065,656 $< L^\star$); the bound had stabilized in the $1.62–$1.63M band by iteration 25.

6 Policy Classes

The v0.3 policy set comprises simple baselines (`reject_all`, `accept_all_feasible`, `myopic_margin`, `bid_price`), a dependency-free linear surrogate (`surrogate_linear`), a finite rollout teacher (`rollout_teacher`), and a parametric cascade (`cascade_surrogate_rollout`). The simple baselines are defined as in Kim et al. [2004]-style truckload acceptance and serve as lower-effort anchors. The rollout teacher implements Bertsekas et al. [1997]-style finite-lookahead Monte Carlo expansion with common-random-number accept/reject branches.

6.1 Surrogate

The linear surrogate fits the rollout-label dataset $\{(x_i, y_i)\}_{i=1}^{N}$ where y_i is the rollout teacher's expected incremental value for decision i and x_i is a feature vector including load attributes, fleet-state features, feasibility-probe features (`feasible_accept`, `service_failure_risk`, `realized_profit_if_feasible`), terminal-value features (`terminal_origin_value`, `terminal_destination_value`, `terminal_delta`), and temporal price features (`price_wave_multiplier`, `price_window_premium`). The fit is a closed-form ridge regression; no third-party dependencies are required. A surrogate-only decision additionally consults the feasibility map and rejects if the copied-fleet probe fails, preventing the surrogate from paying gratuitous L1 penalties.

Features are standardized to zero mean and unit variance on the training labels (a feature with zero training variance is left at unit scale), and an explicit, unregularized bias term is prepended; the target y_i is the rollout incremental-value label divided by a fixed scale constant. Weights solve the ridge normal equations $(X^\top X + \lambda I)\, w = X^\top y$ directly with $\lambda = 0.25$ applied to every weight except the bias. Labels are generated by the rollout teacher on the train-seed streams; all reported surrogate and cascade numbers are evaluated on disjoint eval seeds, and the held-out fit is reported in `freightbidbench_static_label_fit.csv`.

6.2 Cascade

Let π_θ^S denote the surrogate policy of Section 6.1 (including its feasibility guard), let $\hat{V}_\theta(s, a)$ denote its score for action a in state s, and let $\Delta_\theta(s) := \hat{V}_\theta(s, 1) - \hat{V}_\theta(s, 0)$ be the signed score margin. The cascade has two escalation triggers. The *boundary* trigger escalates decisions within a dollar band of the surrogate's accept/reject boundary; the *scarcity* trigger escalates decisions for which the origin market has few immediately available trucks. Concretely, let $o(s)$ denote the candidate load's origin market in state s and let

$$n_o(s) := \big|\big\{k : \ell_t^{(k)} = o(s),\ \tau_t^{(k)} \le t\big\}\big| \tag{14}$$

denote the count of trucks in $o(s)$ that are immediately available at the decision time. The cascade policy with boundary band $\beta \ge 0$ and scarcity threshold $\kappa \in \{-1\} \cup \mathbb{Z}_{\ge 0}$ (with $\kappa = -1$ disabling the scarcity trigger) is

$$\pi_\theta^{\beta,\kappa}(s) = \begin{cases} \pi^R(s) & \text{if } E(s; \beta, \kappa) = 1, \\ \pi_\theta^S(s) & \text{otherwise,} \end{cases} \tag{15}$$

where the escalation indicator is

$$E(s; \beta, \kappa) := \mathbf{1}\big\{|\Delta_\theta(s)| \le \beta\big\} \ \vee\ \mathbf{1}\big\{n_o(s) \le \kappa\big\}. \tag{16}$$

Either trigger alone suffices to escalate. The scarcity trigger intuitively defers the highest-stakes capacity decisions to the rollout teacher: when the origin market is near-empty, the surrogate's bias on the disagreement set is most consequential and the value of an accurate decision is highest. The cascade has three parameters: the surrogate θ, the boundary band β, and the scarcity threshold κ. The released configuration freezes $\kappa = 2$ and sweeps $\beta \in \{0, 250, 500, 700, 900\}$, reporting the full frontier and a representative band $\beta = \$500$.

Proposition 4 (Cascade limit behaviour)**.** *Suppose $\Delta_\theta(s)$ has no atom at 0 under the stationary state distribution induced by π_θ^S (i.e. $\mathbb{P}(\Delta_\theta(s) = 0) = 0$; Δ_θ may otherwise have atoms, as it does when features are discrete), and let*

$$\varphi(\beta, \kappa) := \mathbb{P}\big(E(s; \beta, \kappa) = 1\big) \tag{17}$$

denote the cascade's rollout-call share. Then:

(a) With the scarcity trigger disabled and $\beta = 0$, $\pi_\theta^{0,-1} = \pi_\theta^S$ almost surely: the only escalation set is $\{\Delta_\theta = 0\}$, which is null since Δ_θ has no atom at 0. With the scarcity trigger active ($\kappa \geq 0$) the cascade escalates additionally on $\{n_o \leq \kappa\}$; this set need not be null—empty origin markets occur, notably on `scarce`*—so the cascade does* not *reduce to the pure surrogate at $\kappa = 0$.*

(b) $\lim_{\beta\to\infty} \pi_\theta^{\beta,\kappa} = \pi^R$ pointwise on the support of Δ_θ, for any $\kappa \geq 0$. Similarly, $\pi_\theta^{\beta,K} = \pi^R$ for $\kappa = K$ at least as large as the fleet size, regardless of β.

(c) $\varphi(\beta, \kappa)$ is non-decreasing in β (holding κ fixed) and in κ (holding β fixed); it is right-continuous in β and continuous at every β that is not an atom of $|\Delta_\theta|$.

(d) The expected decision latency $L(\beta, \kappa) = \big(1 - \varphi(\beta, \kappa)\big)L_S + \varphi(\beta, \kappa)\, L_R$ is non-decreasing in both arguments, where L_S and L_R are the surrogate and rollout per-decision latencies and $L_R \geq L_S$.

Sketch. For (a): with the scarcity trigger disabled the only escalation set is $\{\Delta_\theta = 0\}$, which is null because Δ_θ has no atom at 0; hence the cascade follows π_θ^S almost surely. When $\kappa \geq 0$ the additional escalation set $\{n_o \leq \kappa\}$ carries positive probability in general, so the reduction fails. For (b): for any state s, there exists B such that $|\Delta_\theta(s)| \leq B$, hence $\beta \geq B$ forces $E(s; \beta, \kappa) = 1$; the second statement follows because $n_o \leq K$ trivially. Statement (c) follows from the union form of E and the fact that increasing either threshold weakly enlarges the escalation set. Statement (d) follows from (c) and $L_R \geq L_S$. □

Proposition 4(d) gives the cascade its operational meaning: (β, κ) is a two-knob tradeoff between decision latency and agreement with the rollout teacher. Fixing $\kappa = 2$ and sweeping β, as in the released configuration, traces a one-dimensional frontier within the larger (β, κ) family. The empirical frontier in Section 8 characterizes the profit side of the tradeoff per scenario.

7 Computational Considerations

State space. The fleet state F_t is a K-tuple of (ℓ, τ, h, d) truck records. With $|\mathcal{S}| = 12$ markets, fleet size $K \leq 90$, and discretized continuous time, the raw state space is too large for exact enumeration. The exact small-prefix DP is tractable because the prefix length L is small and the deterministic post-acceptance transition map allows fleet-hash memoization.

Exact DP scaling. The exact DP search evaluates up to 2^L realizations of the accept/reject tree. With fleet-hash memoization the effective state count is much smaller; empirically the search evaluates $\sim$ 8k states for $L = 12$ on `tight`. Doubling L approximately doubles wall-clock time but increases peak memory by an order of magnitude in the worst case. We cap $L = 16$ in the released configuration.

Rollout teacher. Each rollout decision expands accept and reject branches under common random numbers for the remaining label budget. Per-decision cost scales linearly in branch depth and label width; on a single thread, mean rollout latency is on the order of 20–30 ms per decision on the v0.3 scenarios.

Table 1: Layer marginal effects on the calibration-gate policies (three-seed full-horizon sweeps). Each row varies one layer with the other layers held at the pre-layer baseline. The gate criterion encodes the predicted class separation; "met" means all policy comparisons in the criterion pass at the listed parameter value.

Layer	Calibration gate	Tight	Scarce
L1 ($\rho \in \{0, 10\}$)	`myopic`, `bid_price` < `accept_all_feasible`	met at $\rho = 10$ (gap −\$3.1k)	met at $\rho = 10$ (gap −\$3.7k)
L2 ($\omega \in \{0, 0.25\}$, L1 on)	`accept_all_feasible` retention ≤ 95%	met at $\omega = 0.25$ (89.7%)	met at $\omega = 0.25$ (93.9%)
L3 (amp $\in \{0, 0.5\}$, L1+L2 on)	best-simple retention gap ≥ 10 pp	met at amp = 0.5 (11.1 pp)	met at amp = 0.5 (14.5 pp)

Cascade. The surrogate handles all decisions in $O(d)$ time where d is the feature dimension; only escalated decisions invoke the rollout teacher. The escalation indicator $E(s; \beta, \kappa)$ costs $O(1)$ per decision (a count of available trucks in the origin market plus a single threshold comparison on $|\Delta_\theta|$). The expected per-decision cost is $(1 - \varphi(\beta, \kappa))\, c_S + \varphi(\beta, \kappa)\, c_R$ with $c_R \gg c_S$, matching Proposition 4(d).

8 Experiments

8.1 Setup

All experiments use the v0.3 scenario contract `scenario-v0.3.2`, default first seed `20260506`, and the public-calibrated lane table at `data/processed/v1_usda_faf_mapped_lanes.csv`. Reported numbers are computed from the runs in `benchmark_runs/v03_sweeps/` and assembled into `benchmark_runs/paper_v03/`. We focus on the `tight` and `scarce` scenarios, where v0.3 economics bite; the `mild` scenario remains effectively flat under v0.3 and is reported as a negative control in the manifest. The headline methods table reports ten paired (train, eval) seeds; paired-bootstrap 95% confidence intervals and policy-vs-rollout deltas are computed by `scripts/analyze_policy_deltas.py` (20,000 resamples).

8.2 Layer Ablation

We isolate the contribution of each v0.3 reward layer by holding the remaining layers at their pre-layer baseline values and varying only the layer in question. The calibration gate for each layer encodes the predicted policy-class separation: L1 should push feasibility-blind policies below the feasibility-aware greedy baseline; L2 should reduce `accept_all_feasible` retention below 95% of rollout; and L3 should widen the best-simple retention gap to at least 10 percentage points below rollout. Table 1 summarizes the three-seed full-horizon sweep results assembled from `reports/service_failure_penalty_sweep_report.md`, `reports/terminal_value_sweep_report.md`, and `reports/demand_wave_sweep_report.md`.

Three patterns confirm Proposition 1 and the qualitative role of L2 and L3.

- **L1 separates feasibility-blind policies.** Under $\rho = \$10$, `myopic_margin` and `bid_price` fall strictly below `accept_all_feasible` on both scenarios. The realized gap (−\$3.1k tight, −\$3.7k scarce) matches the infeasible-accept counts multiplied by ρ, in agreement with Proposition 1.
- **L2 demotes `accept_all_feasible`.** Terminal value weight $\omega = 0.25$ alone reduces `accept_all_feasible` retention to 89.7% on tight and 93.9% on scarce. L2 acts on the feasibility-aware greedy policy exactly because that policy is otherwise insensitive to terminal fleet positioning.

Table 2: Ten-seed methods comparison under `scenario-v0.3.2`. Retention is mean closed-loop profit relative to the rollout teacher. The cascade row uses a single *common* representative band $\beta = \$500$ for both scenarios at frozen scarcity threshold $\kappa = 2$—not the per-scenario best frontier point (the full frontier in Table 3 peaks at a higher band on `scarce`). The last column is the paired-bootstrap 95% CI on the cascade-minus-rollout profit delta ($n = 10$); an interval containing \$0 means the cascade is statistically indistinguishable from rollout.

Scenario	Best simple	Surrogate	Cascade	Cascade ms	Rollout ms	Cascade−Rollout Δ (95% CI)
`tight`	91.0%	94.2%	98.2%	12.95	32.11	−\$23.3k [−\$61.2k, +\$11.6k]
`scarce`	86.5%	89.3%	98.0%	11.54	20.59	−\$20.7k [−\$33.4k, −\$7.2k]
`mild`	100.1%	98.2%	99.7%	8.31	49.50	−\$4.2k [−\$18.2k, +\$11.0k]

Table 3: Cascade latency–profit frontier over the boundary band $\beta \in \{0, 250, 500, 700, 900\}$ at $\kappa = 2$. Rollout share is the fraction of decisions escalated to the rollout teacher.

Scenario	β	Retention	Mean latency ms	Rollout share
`tight`	0	97.7%	6.27	28.6%
`tight`	500	98.2%	12.95	44.7%
`tight`	900	99.0%	17.72	57.0%
`scarce`	0	93.7%	6.09	45.4%
`scarce`	500	98.0%	11.54	59.6%
`scarce`	900	99.5%	15.72	78.6%

- **L3 widens the rollout gap.** The price-premium window at amplitude 0.5 widens the rollout-versus-best-simple gap to 11.1 pp on tight and 14.5 pp on scarce, providing the residual headroom that the cascade results in Section 8.3 exploit.

8.3 Methods Frontier

Table 2 reports the ten-seed methods comparison under the frozen `scenario-v0.3.2` configuration, and Table 3 the cascade latency–profit frontier over the boundary band. The cascade is the only stdlib policy that closes the gap to rollout.

Three interpretive points. First, at $\beta = \$500$ the cascade recovers $\sim 98\%$ of rollout profit on both stress scenarios at 40% (`tight`) to 56% (`scarce`) of rollout's mean decision latency. On `tight` the paired cascade-minus-rollout difference is not statistically significant (the 95% CI spans zero), so the cascade matches the rollout teacher to within sampling error while escalating fewer than half of all decisions; on `scarce` a small but significant $\sim 2\%$ gap remains. Second, the frontier is smooth and monotone in β (Table 3), consistent with Proposition 4(d): the scarcity trigger alone ($\beta = \$0$) already recovers 97.7% on `tight` but only 93.7% on `scarce`, where the surrogate's disagreement set with rollout extends well beyond the scarcity regime. Third, the standalone surrogate clears the best-simple bar on both scenarios (94.2% vs 91.0% on `tight`; 89.3% vs 86.5% on `scarce`), but still leaves 6–11 points of rollout value on the table, all of which the cascade recovers. The surrogate is a competent ranker but a poor substitute for selective lookahead on the high-stakes decisions that the scarcity and boundary triggers escalate.

8.4 Hindsight Diagnostics

Table 4 reports the exact small-prefix dynamic program for $L = 12$ on `tight`. The exact ceiling is matched by simple feasibility-aware policies on this prefix length, confirming that small realized

Table 4: Exact small-prefix hindsight diagnostic on `tight` with $L = 12$ loads.

Scenario	L	Hindsight	States	Best simple	Simple ret.	Rollout ret.
`tight`	12	$53,419	8,191	`accept_all_feasible`	100.0%	90.0%

Table 5: Full-horizon upper bounds and rollout-teacher retention against each ceiling. The Lagrangian bound retains per-truck structure and is 20.7% tighter than the LP relaxation on `tight` and 39.3% tighter on `scarce`; rollout retention against the ceiling rises from 53.6%/39.8% (LP) to 67.6%/65.7% (Lagrangian).

		LP-style		Lagrangian-per-truck		
Scenario	Loads	Bound	Rollout ret.	Bound	Rollout ret.	Tightening
`tight`	995	$2,377,500	53.6%	$1,885,043	67.6%	20.7%
`scarce`	1,154	$2,675,525	39.8%	$1,623,084	65.7%	39.3%

prefixes are flat for v0.3 economics and reinforcing that exact DP is a diagnostic rather than a paper-scale ceiling.

Table 5 compares the LP-style and Lagrangian-per-truck upper bounds on `tight` and `scarce`. Both are valid upper bounds by Propositions 2 and 3; the Lagrangian retains per-truck HOS, location, and sequencing structure while only the cross-truck assignment constraint is dualized, so it is substantially tighter.

The looseness reduction is structural rather than numerical: the Lagrangian relaxation keeps per-truck location continuity, sequencing, and HOS budgets while dualizing only the cross-truck assignment. This matches the looseness decomposition in Section 5.2 — sequencing and integrality dominated the LP's slack, and the Lagrangian recovers most of that lost tightness. Rollout retention against the Lagrangian bound rises from 53.6% to 67.6% on `tight` and from 39.8% to 65.7% on `scarce`; the larger tightening on `scarce` (39.3%) reflects that the loose LP bound was most optimistic exactly where capacity is most binding. In both cases the Lagrangian ceiling locates the v0.3 rollout teacher much closer to the dependency-free upper bound and characterizes methodological headroom more honestly.

8.5 Sensitivity Analysis

Sensitivity of the qualitative ordering to the v0.3 parameters is reported in Table 6. For each parameter we sweep a small grid around the frozen value, holding the other parameters at their frozen levels. The ordering of policies is preserved across the swept ranges, supporting the claim that the v0.3 results are not an artifact of the specific calibrated values.

9 Discussion and Managerial Insights

Three observations follow from the v0.3 results that translate beyond the benchmark itself.

Operational feasibility is a reward, not a side diagnostic. Proposition 1 formalizes a simple operational truth: any policy that ranks tenders without checking feasibility will pay a linear service-failure tax on its infeasible-accept rate. In a production context, scoring systems that surface tenders to dispatchers without upstream feasibility checks therefore have a hidden expected cost

Table 6: Sensitivity of the qualitative ordering to v0.3 parameter choices. Each row reports whether the calibration gate (`myopic/bid_price` below `accept_all_feasible` for L1; `accept_all_feasible` retention $\leq 95\%$ for L2; best simple retention $\leq 90\%$ for L3) is met.

Parameter	Tested values	Frozen value	Gate met (tight)	Gate met (scarce)
ρ (L1, \$)	{0, 10, 25, 50}	10	{no, yes, yes, yes}	{no, yes, yes, yes}
ω (L2)	{0, 0.1, 0.25, 0.5, 1.0}	0.25	{no, no, yes, yes, yes}	{no, no, yes, yes, yes}
L3 amplitude	{0, 0.25, 0.5, 0.75}	0.5	{no, no, yes, yes}	{no, yes, yes, yes}

that scales with the operational mismatch rate. The L1 calibration in v0.3 suggests this cost can be substantial even with a modest per-event penalty.

Cascade structure is the right framing, not standalone surrogate. The v0.3 surrogate is biased on the disagreement set with the rollout teacher in the `scarce` regime, where capacity-positioning is most consequential. The cascade recovers nearly all of rollout's profit by escalating exactly these uncertain decisions. The implication for practice is that a cheap learned model deployed as a sole policy will under-perform in the regimes where future-aware decisions matter most; the value of the model comes from selectively deferring uncertain decisions to a more expensive teacher.

Methodological headroom is smaller than the LP relaxation suggests, once per-truck structure is respected. The LP-style bound loses most of its tightness to dropping sequencing, location, and integrality constraints simultaneously; the Lagrangian-per-truck information relaxation (Proposition 3) retains those constraints and tightens the bound by 20.7% on `tight` and 39.3% on `scarce`. Rollout retention against the Lagrangian bound rises from 53.6% to 67.6% on `tight` and from 39.8% to 65.7% on `scarce`, characterizing the achievable methodological headroom more honestly. The remaining gap to the Lagrangian bound is dominated by the residual cross-truck-assignment dual slack: the headroom for future methods work lies in tighter inter-truck coordination (joint-decision lookahead, dual-price-aware features in the surrogate, or anytime joint optimization), not in further refinement of single- tender scoring.

10 Limitations

FreightBidBench remains synthetic and public-calibrated, not a private tender dataset; the calibration is validated against its FAF/USDA sources in Appendix B, which also documents that the v1 USDA-reefer lane subset concentrates load draws on a few high-volume Texas and Georgia origins. The HOS model is simplified to property-carrying 11/14/10 clocks and omits split-sleeper, recap, home-time, and team-driver rules. The rollout teacher is a finite-lookahead stochastic benchmark, not a true oracle: a cheaper policy can exceed 100% retention on a given seed. The exact hindsight DP is tractable only for small load prefixes. The relaxed full-horizon bound is intentionally loose; it characterizes problem hardness rather than achievable profit. The dependency-free methods track means that stronger learned baselines (gradient-boosted trees, neural surrogates) are outside scope for v0.3 and should not be required for reproducibility tests.

11 Conclusion

FreightBidBench v0.3 turns the v0.2 feasibility calibration into a sharper, reproducible benchmark for online truckload bid acceptance. The three new reward components each separate a distinct policy class (Section 4); the two-tier hindsight ceilings (Section 5) provide both a trustworthy small-instance reference and a structurally honest full-horizon upper bound; and the parametric surrogate-rollout cascade (Section 6) admits a clean limit characterization (Proposition 4) and demonstrates a non-trivial latency-profit frontier on the benchmark (Section 8). The benchmark itself is the primary artifact: future methods work — stronger surrogates, joint-decision lookahead, learned cascade-band selection — can be evaluated against the released ceilings and the versioned scenario contract without ambiguity over what exactly is being compared.

Acknowledgements

The author thanks the open data communities at the U.S. Bureau of Transportation Statistics, the U.S. Department of Agriculture Agricultural Marketing Service, and the Federal Motor Carrier Safety Administration, whose public datasets and regulatory summaries make FreightBidBench possible without proprietary data.

Declaration of Competing Interest

The author is employed by Bubba AI, which develops AI-based load-planning and carrier-operations products in the freight domain addressed by this paper. This study uses only public Freight Analysis Framework and USDA data and a dependency-free, open-source benchmark; no proprietary data or systems were used, and Bubba AI had no role in the study design, the analysis, or the decision to publish. The author declares no other competing interests.

A Reproducibility

Software. Python 3.10 or newer, standard library only at runtime. No third-party dependencies are required to reproduce any table in this paper.

Hardware. Reported single-threaded latencies were measured on a 2024-class laptop (Apple silicon). Closed-loop simulator runtime is dominated by Python interpreter overhead; absolute latencies are reference latencies and not optimized serving latencies.

Versioning. The release pins three version strings in `configs/freightbidbench_v03_scenarios.json`: benchmark version `freightbidbench-v0.3`, scenario configuration version `scenario-v0.3.2`, and policy set version `policy-set-v0.3.0`. The default first seed is `20260506`.

Manifest schema. Every run writes a `freightbidbench_manifest.json` containing the command line, the three version strings, the seed pairs evaluated, the source input paths, the feasibility configuration, the scenarios, the output paths, and the row counts for each output CSV. The manifest is the canonical reproducibility anchor: two runs producing matching manifests must produce matching output CSVs.

Headline run commands. The exact commands used to assemble the tables in Section 8 are:

```
python3 scripts/run_freightbidbench.py \
  --config configs/freightbidbench_v03_scenarios.json \
  --preset standard --scenarios mild,tight,scarce \
  --seed-count 10 --label-limit 200 \
  --cascade-bands 0,250,500,700,900 \
  --output-dir benchmark_runs/v03_sweeps/methods_cascade_seed10_label200
python3 scripts/analyze_policy_deltas.py \
  --run-dir benchmark_runs/v03_sweeps/methods_cascade_seed10_label200 \
  --cascade-band 500

python3 scripts/run_lagrangian_bound.py \
  --config configs/freightbidbench_v03_scenarios.json \
  --scenario scarce --eval-seed 20260507 \
  --iterations 30 --step-scale 100 \
  --lp-bound-reference 2675525 --rollout-reference 1065656 \
  --output-dir benchmark_runs/lagrangian_bound_scarce_full

make calibration-report
make hindsight-smoke
make paper-v03-tables
```

Layer-ablation source. Layer-ablation values in Table 1 are assembled from the calibration sweeps under `benchmark_runs/v03_sweeps/service_failure_penalty/`, `benchmark_runs/v03_sweeps/terminal_value/`, and `benchmark_runs/v03_sweeps/demand_waves_price_selected_seed3/` as documented in `reports/service_failure_penalty_sweep_report.md`, `reports/terminal_value_sweep_report.md`, and `reports/demand_wave_sweep_report.md`.

B Calibration Validation

The "public-calibrated" claim is not merely nominal: the simulator's load, fleet, distance, price, and terminal-value primitives are all derived from public Freight Analysis Framework (FAF 5.7.1, 2024 truck mode) and USDA Agricultural Marketing Service (AMS Specialty Crops Market News, `fvwtrk` truck-rate) data through the pipeline in `scripts/inspect_public_sources.py`. Candidate loads are drawn with probability proportional to lane `faf_tons_2024`; the initial fleet is placed proportional to per-origin FAF outbound tonnage; lane distance is $1000 \cdot$ `faf_tmiles_2024`/`faf_tons_2024`; posted prices are drawn from each lane's USDA AMS rate band [`rate_low`, `rate_high`]; and the terminal state-value signal $V(\cdot)$ of Equation (6) is computed from FAF outbound intensity and the FAF/USDA net-outbound imbalance panel. The cross-checks below are reproduced by `scripts/analyze_calibration.py` (dependency-free), which writes `reports/calibration_report.md`.

B.1 Origin intensity vs FAF outbound flow. Because load draws and fleet placement are FAF-tonnage-weighted, each origin's load-draw share equals its share of total lane FAF tonnage; FAF outbound and net-outbound tons are the independent cross-check from the state imbalance panel (Table 7).

Table 7: Origin intensity vs FAF outbound flow.

Origin state	Load-draw share	FAF outbound tons 2024	Net outbound tons 2024
Texas	78.8%	1,517,190	+19,720
Georgia	16.3%	348,979	−7,477
California	3.9%	756,331	−1,448
Arizona	0.6%	134,239	−3,447
Washington	0.4%	265,827	−8,130
Colorado	0.1%	157,922	−2,894

B.2 Haul-length distribution. Lane distances span 88–3,081 mi (median 2,211; IQR 1,454–2,770), but the tonnage-weighted mean is only 208 mi: FAF truck tonnage concentrates on short intra-state metro flows. A single lane, Texas→Dallas, accounts for 77.2% of tonnage at 105 mi, and Georgia→Atlanta for 15.0% at 88 mi. The realized closed-loop load mix is therefore short-haul-dominated with a long interstate tail — a faithful reflection of FAF truck flow on this lane subset, and a scope limitation.

B.3 Price calibration vs USDA AMS. Posted prices come from the USDA AMS rate bands (73 of 74 lanes have a positive-width band). The implied per-mile rate has median \$3.28 and IQR \$2.83–\$3.57, consistent with published refrigerated truckload spot rates; the single high outlier is a short, distance-clamped lane.

Scope of the v1 lane set. The USDA-reefer-mapped lane catalog is concentrated on a few high-volume Texas and Georgia origins. This is correct FAF behavior for the mapped subset, but it means the benchmark's closed-loop dynamics are driven substantially by short-haul intra-state repositioning. Broadening lane coverage to the full FAF truck OD matrix is a calibration extension for a future release; operational-metric calibration against ATRI/FMCSA carrier benchmarks is likewise deferred.

References

Daniel Adelman and Adam J. Mersereau. Relaxations of weakly coupled stochastic dynamic programs. *Operations Research*, 56(3):712–727, 2008. doi: 10.1287/opre.1070.0445.

Dimitri P. Bertsekas, John N. Tsitsiklis, and Cynara Wu. Rollout algorithms for combinatorial optimization. *Journal of Heuristics*, 3(3):245–262, 1997. doi: 10.1023/A:1009635226865.

Mark Boddy and Thomas L. Dean. Solving time-dependent planning problems. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI)*, 1989.

David B. Brown, James E. Smith, and Peng Sun. Information relaxations and duality in stochastic dynamic programs. *Operations Research*, 58(4):785–801, 2010. doi: 10.1287/opre.1090.0796.

Justin C. Goodson, Barrett W. Thomas, and Jeffrey W. Ohlmann. A rollout algorithm framework for heuristic solutions to finite-horizon stochastic dynamic programs. *European Journal of Operational Research*, 258(1):216–229, 2017. doi: 10.1016/j.ejor.2016.09.040.

Ahmed Heakl, Yahia Salaheldin Shaaban, Salem Lahlou, Martin Takáč, and Zangir Iklassov. SVRPBench: A realistic benchmark for stochastic vehicle routing problem. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. OpenReview.

Jörg Homberger and Hermann Gehring. A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *European Journal of Operational Research*, 162(1):220–238, 2005. doi: 10.1016/j.ejor.2004.01.027.

Eric J. Horvitz. Reasoning under varying and uncertain resource constraints. In *Proceedings of the 7th National Conference on Artificial Intelligence (AAAI)*, 1988.

Yongjin Kim, Hani S. Mahmassani, and Patrick Jaillet. Dynamic truckload routing, scheduling, and load acceptance for large fleet operation with priority demands. *Transportation Research Record*, 1882(1):120–128, 2004. doi: 10.3141/1882-15.

Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *International Conference on Learning Representations*, 2019.

Mohammadreza Nazari, Afshin Oroojlooy, Lawrence Snyder, and Martin Takáč. Reinforcement learning for solving the vehicle routing problem. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Rahul Mihir Patel, Justin Dumouchelle, Elias B. Khalil, and Merve Bodur. Neur2sp: Neural two-stage stochastic programming. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Victor Pillac, Michel Gendreau, Christelle Guéret, and Andrés L. Medaglia. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11, 2013. doi: 10.1016/j.ejor.2012.08.015.

Warren B. Powell, Hugo P. Simão, and Belgacem Bouzaiene-Ayari. Approximate dynamic programming in transportation and logistics: A unified framework. *EURO Journal on Transportation and Logistics*, 1(3):237–284, 2012. doi: 10.1007/s13676-012-0015-8.

Warren B. Powell, Belgacem Bouzaiene-Ayari, Coleman Lawrence, Clark Cheng, Shyam Das, and Ricardo Fiorillo. Locomotive planning at norfolk southern using approximate dynamic programming. *Interfaces*, 44(6):567–578, 2014. doi: 10.1287/inte.2014.0764.

Ulrike M. Ritzinger, Jakob Puchinger, and Richard F. Hartl. A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*, 54(1):215–231, 2016. doi: 10.1080/00207543.2015.1043403.

Stuart Russell and Eric Wefald. Principles of metareasoning. *Artificial Intelligence*, 49(1–3):361–395, 1991.

Nicola Secomandi and François Margot. Reoptimization approaches for the vehicle-routing problem with stochastic demands. *Operations Research*, 57(1):214–230, 2009. doi: 10.1287/opre.1080.0520.

Hugo P. Simão, Joseph Day, Abraham P. George, Ted Gifford, John Nienow, and Warren B. Powell. An approximate dynamic programming algorithm for large-scale fleet management: A case application. *Transportation Science*, 43(2):178–197, 2009. doi: 10.1287/trsc.1080.0238.

Marius M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987. doi: 10.1287/opre.35.2.254.

Daniel Tjokroamidjojo, Erhan Kutanoglu, and G. Don Taylor. Quantifying the value of advance load information in truckload trucking. *Transportation Research Part E: Logistics and Transportation Review*, 42(4):340–357, 2006. doi: 10.1016/j.tre.2005.04.001.

Huseyin Topaloglu and Warren B. Powell. Dynamic-programming approximations for stochastic time-staged integer multicommodity-flow problems. *INFORMS Journal on Computing*, 18(1): 31–42, 2006. doi: 10.1287/ijoc.1040.0079.

Eduardo Uchoa, Diego Pecin, Artur Pessoa, Marcus Poggi, Thibaut Vidal, and Anand Subramanian. New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, 257(3):845–858, 2017. doi: 10.1016/j.ejor.2016.08.012.

Marlin W. Ulmer and Barrett W. Thomas. Meso-parametric value function approximation for dynamic customer acceptances in delivery routing. *European Journal of Operational Research*, 285(1):183–195, 2020. doi: 10.1016/j.ejor.2019.04.029.

Marlin W. Ulmer, Justin C. Goodson, Dirk C. Mattfeld, and Marco Hennig. Offline–online approximate dynamic programming for dynamic vehicle routing with stochastic requests. *Transportation Science*, 53(1):185–202, 2019. doi: 10.1287/trsc.2017.0767.

U.S. Department of Agriculture, Agricultural Marketing Service. Specialty crops national truck rate report (fvwtrk), 2026. Market News report, report slug 2375.

U.S. Department of Transportation, Bureau of Transportation Statistics and Federal Highway Administration. Freight analysis framework, faf5, 2017. Dataset.

Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

Jian Yang, Patrick Jaillet, and Hani Mahmassani. Real-time multivehicle truckload pickup and delivery problems. *Transportation Science*, 38(2):135–148, 2004. doi: 10.1287/trsc.1030.0068.

Shlomo Zilberstein. Using anytime algorithms in intelligent systems. *AI Magazine*, 17(3):73–83, 1996.