AUDITABLE REGRET FRONTIERS FOR ONLINE ROUTING WITH ENDOGENOUS FEASIBLE SETS

Anonymous authors
Paper under double-blind review

ABSTRACT

Online routing consumes resources that change which routes remain feasible later. A deployment log can therefore support comparisons among actions on visited states without reconstructing the state trajectory that another policy would have generated. We formalize this distinction through an *auditable regret frontier* with two axes. L0/L1/L2 describe executed-action feedback, same-state candidate audit, and counterfactual-state reconstruction. Conditional on L1, a repayment window W controls how long a comparator may carry positive safety pressure before repaying it. We show that unrestricted L1 comparators admit instances in which no algorithm can guarantee both sublinear reward regret and sublinear violation. In a normalized exact decision-time L1 oracle game with strict fallback, the minimax weighted cost is 0 at $W = 1$ and $\Theta(\sqrt{WT})$ for $2 \le W \le T$. A CVQ-Routing specialization with exact coefficients supplies the matching-order upper bound. The deployable controller instead uses surrogates, and its pathwise certificate records coefficient mismatch, route-search error, queue drift, and comparator debt. Controlled synthetic and public-data replays probe these terms and the difference between same-state L1 evaluation and own-trajectory L2 replay. The resulting framework makes explicit which comparison objects are supported by an audit interface and how comparator strength changes their attainable cost.

1 INTRODUCTION

Routing spends resources that determine which routes remain feasible later. Inventory routing and service chaining have this feedback through residual capacity (Coelho et al., 2014; Medhat et al., 2017); trusted-relay quantum key distribution (QKD) has it through the key material consumed along a path (Peev et al., 2009; Sasaki et al., 2011; López et al., 2026; Hao et al., 2025). Before evaluating an alternative rule from a trace, an operator must therefore ask: *which comparison does the recorded information actually support?*

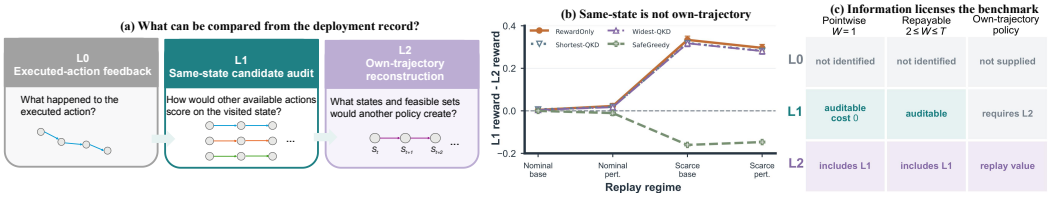

Figure 1: **The record determines what can be compared; the comparator determines the remaining cost.** (a) Actions change future route availability. (b) L1 minus L2 reward from the existing replay: all four target rules and all four regimes, with mean $\pm 1.96 s/\sqrt{12}$ across paired weeks; line segments are visual connectors across categorical settings. Shortest/Widest abbreviate the QKD controls. (c) Categorical audit map; costs apply to the normalized exact decision-time L1 game. Panels (a,c) are schematic.

This question has two parts. First, exact candidate values on visited states do not reconstruct the states generated by a different action history. Second, auditability alone does not make a comparator

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

compatible with the learner's safety budget. We therefore organize the problem along two axes: an information axis—L0 executed outcomes, L1 validated candidate values on realized states, and L2 counterfactual-state reconstruction—and a comparator axis, where the repayment window W controls how long an L1 comparator may carry positive pressure before repaying it. We call the resulting object an *auditable regret frontier*.

Existing tools address pieces of these axes. Virtual queues and primal–dual methods control long-run violations (Neely, 2010; Liu et al., 2023); policy regret handles alternative action histories (Arora et al., 2012; Bhatia & Sridharan, 2020); OPE studies policy-value identification from logged data (Uehara et al., 2026; Khan et al., 2024); and window-budget benchmarks restrict repayment horizons (Liakopoulos et al., 2019; Braverman et al., 2025). Our theory connects these pieces: L0 may leave same-state regret unidentified, L1 identifies realized-state candidate values, L2 supplies the additional state reconstruction needed for own-trajectory comparison, and conditional on L1 the W-window restriction yields exact-L1 minimax cost 0 at $W = 1$ and $\Theta(\sqrt{WT})$ for $2 \le W \le T$.

The decision-time exact-L1 game is used only as an oracle benchmark for allocation uncertainty. Deployable CVQ-Routing acts on surrogates and uses post-hoc L1 validation to audit coefficient and search error. The experiments separately vary repayment windows, approximation error, and state reconstruction.

The theory separates an information question from a comparator question. At the information stage, L0 can leave same-state regret unidentified, while L1 identifies candidate values only on states that were actually visited; L2 supplies the additional state reconstruction needed for an own-trajectory policy comparison. Conditional on L1, unrestricted comparators admit a linear safety–regret obstruction. Imposing W-window repayment yields normalized exact-L1 minimax cost 0 at $W = 1$ and $\Theta(\sqrt{WT})$ for $2 \le W \le T$.

The exact decision-time L1 game is an oracle benchmark: it reveals all current candidate rewards and pressures before the action is chosen. Its purpose is to isolate future allocation uncertainty from current-value estimation. Deployable CVQ-Routing instead acts on surrogates. It retains the realized candidate set and uses post-hoc L1 validation to measure coefficient mismatch and route-search loss. The pathwise certificate keeps these approximation channels separate from the minimax benchmark.

Each experiment is tied to a theoretical object. Exact window comparators probe the W-dependent guarantees; forecast and route-search sweeps measure the approximation charges in the pathwise certificate; and reset replay illustrates the L1–L2 state-alignment term using Abilene traffic and measured QKD secure-key-rate inputs.

Contributions. **(i)** We formalize the L0/L1/L2 audit hierarchy and isolate the state-alignment term separating same-state from own-trajectory evaluation. **(ii)** We characterize the comparator-strength axis through the exact-L1 $\Theta(\sqrt{WT})$ frontier and its distinct $W = 1$ endpoint. **(iii)** We develop an executable routing controller whose pathwise certificate separates queue drift, comparator debt, coefficient mismatch, and route-search error.

2 Related Work

We organize the closest literature into three strands: (1) constrained online learning and repayment-window benchmarks, (2) policy regret, stateful dynamics, and off-policy evaluation, and (3) resource-aware routing and QKD networks. These strands address different parts of our problem: online safety control, counterfactual comparison, and endogenous resource dynamics, respectively.

(1) Constrained online learning and repayment windows. Virtual-queue and primal–dual methods provide the basic machinery for controlling long-run constraint violations (Neely, 2010; Liu et al., 2023; Kolev et al., 2023; Stradi et al., 2024), with extensions to adaptive constraints (Castiglioni et al., 2024; Bernasconi et al., 2024a; 2025). A complementary line of work restricts comparator resource use over finite windows: Liakopoulos et al. (2019) and Braverman et al. (2025) formalize benchmarks in which temporary resource imbalance must be repaid within a prescribed horizon. Our formulation separates this comparator restriction from the information available to evaluate it: the audit level determines which alternative values are supported by the record, while the repayment window W determines the strength of the realized-state comparator. **(2) Policy regret, stateful**

dynamics, and off-policy evaluation. Policy regret and online learning with memory account for settings in which earlier actions change later losses, states, or feasible decisions (Arora et al., 2012; Anava et al., 2015; Hall & Willett, 2013; Kumar et al., 2023; Bhatia & Sridharan, 2020; Liu et al., 2026). Off-policy evaluation instead asks when the value of an alternative policy can be identified from logged data under overlap or structural assumptions (Uehara et al., 2026; Khan et al., 2024). These ideas correspond to different information requirements in our setting. L1 evaluates alternative actions on the states actually visited by the learner, whereas an own-trajectory policy comparison additionally requires reconstruction of the states and feasible sets induced by a different action history, which is the role of L2. **(3) Resource-aware and QKD routing.** Resource-aware learning, inventory routing, and network-routing methods study decision making under capacity or resource constraints (Bernasconi et al., 2024b; Ma et al., 2024; Cesa-Bianchi & Lugosi, 2012; Awerbuch & Kleinberg, 2004; Fortz & Thorup, 2000; Coelho et al., 2014). Trusted-relay QKD routing additionally couples routing decisions to residual key resources (López et al., 2026; Hao et al., 2025). These models motivate our endogenous-feasibility setting, where an executed route changes the resources and routes available later. Our focus is the associated evaluation problem: which regret or policy-value comparisons are supported by the trace generated by such a controller? Appendix B provides a representative-work matrix that locates these formulations along the information and comparator axes.

3 Auditable Information and Comparator Frontier

3.1 Endogenous Feasibility and the Recorded Information

Two terms recur throughout the theory. We call g_t *pressure*: positive coordinates consume long-run safety budget, while negative coordinates repay it. We call an evaluation *same-state* when it compares actions on the learner's realized state, and *own-trajectory* when it evaluates a policy on the states generated by that policy's own action history.

Let $G = (V, E)$ be a directed edge universe, $m = |E|$, and $q_t \in [0, \bar{q}]^m$ the pre-decision inventory. A request $\omega_t = (s_t, d_t)$ arrives with availability E_t^{av}, eligibility/trust filter τ_t, consumption $\nu_t \succeq 0$, and replenishment $a_t \succeq 0$. The operational edges and inventory evolve as

$$E_t^{\mathrm{op}} = \{e \in E_t^{\mathrm{av}} : e \text{ passes } \tau_t,\ q_{e,t} \geq \nu_{e,t}\}, \qquad q_{t+1} = \Pi_{[0,\bar{q}]}(q_t - \mathrm{diag}(\nu_t)x_t + a_t). \tag{1}$$

Fix a declared route catalog $\mathcal{C}(\omega_t)$ for request ω_t; it may contain all simple s_t–d_t paths. Let

$$\mathcal{P}_t = \{p \in \mathcal{C}(\omega_t) : p \subseteq G_t^{\mathrm{op}}\}, \qquad \mathcal{X}_t^+ = \{0\} \cup \{\chi(p) : p \in \mathcal{P}_t\},$$

where $\chi(p)$ is path incidence and 0 is fallback. The realized state is $S_t = (q_t, G_t^{\mathrm{op}}, \omega_t)$. Candidate reward is $r_t(x)$; positive long-run pressure $g_t(x, q_t) \in \mathbb{R}^K$ spends safety budget and negative pressure repays it. All realized-state comparators and certificates below are relative to this declared candidate universe. The implementation uses

$$g_t(x, q_t) = M_t(q_t)x - b_t, \qquad b_t \succeq 0, \tag{2}$$

Thus fallback is weakly safe: $g_t(0, q_t) = -b_t \preceq 0$. Current feasibility is a physical inventory constraint; cumulative pressure is a separate safety criterion (Coelho et al., 2014; López et al., 2026; Hao et al., 2025).

Definition 1 (Auditability levels)**.** ***L0*** *records S_t, the executed x_t, and feedback for x_t.* ***L1*** *additionally records or reconstructs the realized candidate set $\mathcal{X}_t^+$ and validated post-hoc reward/safety values for its candidates, either from logged telemetry or from a specified evaluator.* ***L2*** *additionally reconstructs the states, feasible candidate sets, and corresponding reward/safety values induced by an unexecuted policy, for example through resets, shadow replay, or a validated structural simulator/evaluator.*

For additive rewards $r_t(x) = \theta_t^\top x$, let A_t stack candidate incidence vectors. Edge-level telemetry $(A_t, \theta_t, M_t(q_t), b_t)$ reconstructs all realized candidate values as $A_t\theta_t$ and $A_t M_t(q_t)^\top - \mathbf{1}b_t^\top$ (Proposition 1). Thus L1 can be supplied without physically executing each route. Execution-only latent outcomes remain an action-support issue, while values on unvisited states require the separate L2 interface.

3.2 From Same-State Outcomes to Alternative Trajectories

Without candidate-outcome support, even the same-state comparison can be unidentified (Uehara et al., 2026; Khan et al., 2024).

Proposition 2 (L0 non-identification)**.** *There exist two one-round environments producing the same L0 record for the executed action but different regret to the best feasible same-state action. Hence exact same-state regret is not identified by executed-action feedback alone.*

The proof is in Appendix F. L1 still compares actions only on the learner's states. A target policy π instead visits S_t^π, whereas the learner A visits S_t^A. Write V_T for total reward on each rule's own trajectory and PolReg_T for the resulting hindsight-policy reward gap. Future resource preservation can make this online comparison hard even when current outcomes are known exactly.

Theorem 1 (Counterfactual policy-regret lower bound)**.** *For every even $T = 2n$, there is a two-edge endogenous-inventory family of deterministic instances such that every randomized online algorithm A satisfies*

$$\sup_{\omega} \mathbb{E}[\mathrm{PolReg}_T(A;\omega)] \geq T/4.$$

Both edges initially reward service, but only one remains useful later. Hindsight preserves that edge; the learner must act before learning which edge it is (Figure 2; Appendix H). Thus Theorem 1 isolates online uncertainty about future resource value.

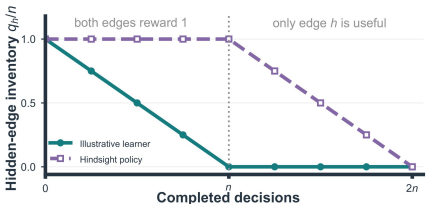

Figure 2: **Equal immediate rewards can lead to different future feasible sets.** A schematic trajectory from the two-edge construction in Theorem 1, with $T = 8$ and $n = 4$. The displayed learner spends edge h early, whereas hindsight preserves it. L1 compares actions at S_t^A; L2 reconstructs S_t^π.

Retrospective evaluation raises a separate question: what additional information is needed to compare against a policy on its own trajectory? Let $x_t^\pi = \pi(S_t^\pi)$ and let r_t^π denote reward on π's own trajectory.

Proposition 3 (State-alignment decomposition)**.** *If there exists an action $\tilde{x}_t^\pi$ that is pointwise safe on the learner-realized state S_t^A and satisfies $|r_t^\pi(x_t^\pi) - r_t(\tilde{x}_t^\pi)| \leq \epsilon_t^\pi$, then*

$$V_T(\pi) - V_T(A) \leq \sum_{t=1}^{T} \big(r_t(x_t^\star) - r_t(x_t)\big) + \sum_{t=1}^{T} \epsilon_t^\pi,$$

where $x_t^\star$ is the best pointwise-safe action on S_t^A.

The first term is the pointwise-safe regret $\mathrm{Reg}_T^{\mathrm{tc}}$ defined in Equation 4; the proof is in Appendix I. The second term is the state-alignment charge between the policy's own trajectory and an aligned action on the learner's realized state. This makes the L1–L2 boundary concrete: reset evaluation, shadow replay, or a validated transition model can be used to estimate or bound the alignment charge.

3.3 Comparator Safety and the Exact-L1 Game

Within L1, a comparator $y_{1:T}$ uses $y_t \in \mathcal{X}_t^+$ on learner-realized states. Let $\lambda_t \in \mathbb{R}_+^K$ be the learner's logged virtual queue, a nonnegative shadow price accumulated from realized pressure; its update is introduced in Section 4. Define

$$\mathrm{Reg}_T(y) = \sum_{t=1}^{T} \big(r_t(y_t) - r_t(x_t)\big), \qquad \mathcal{D}_T(y) = \sum_{t=1}^{T} \lambda_t^\top g_t(y_t, q_t). \tag{3}$$

The first quantity is the auditable reward gap; the second prices comparator pressure using the same logged queue. Pointwise-safe regret and learner violation are

$$x_t^\star \in \arg\max_{x \in \mathcal{X}_t^+ : g_t(x,q_t) \preceq 0} r_t(x), \quad \mathrm{Reg}_T^{\mathrm{tc}} = \mathrm{Reg}_T(x^\star), \quad \mathrm{Viol}_T = \left\| \left[\sum_{t=1}^{T} g_t(x_t, q_t) \right]_+ \right\|_2. \tag{4}$$

Auditability and safety compatibility are distinct: a comparator can be fully observable yet violate the learner's admissible safety budget.

Proposition 4 (Unrestricted L1 comparators are incompatible with joint sublinear safety and regret)**.** *There is a fully L1-audited one-state, two-action instance with bounded rewards and pressure such that, for the comparator that always takes the high-reward action, every online algorithm satisfies*

$$\mathrm{Reg}_T(y) + \tfrac{1}{2}\mathrm{Viol}_T \geq T/2.$$

The fully observed counterexample in Appendix G motivates a repayment restriction rather than additional feedback.

Definition 2 (W-window-safe realized-state comparator)**.** *Fix a partition $\mathcal{B}_W = \{B_j\}_{j=1}^J$ of consecutive rounds with $|B_j| \leq W$. A comparator $y_{1:T}$ is W-window-safe if*

$$\sum_{t \in B_j} g_t(y_t, q_t) \preceq 0 \qquad \textit{for every } B_j.$$

The reward-maximizing finite-catalog benchmark in this class maximizes $\sum_t r_t(y_t)$ subject to these block constraints. $W = 1$ is pointwise safety; $W > 1$ permits temporary positive pressure followed by repayment inside the same audit window.

Fallback makes the class nonempty. Coarsening a fixed partition relaxes safety constraints without adding information; the experimental powers-of-two grid has this nesting property. Arbitrary changes of nonnested block boundaries need not yield nested comparator classes (Liakopoulos et al., 2019; Braverman et al., 2025).

Definition 3 (Exact-L1 decision-time game and minimax cost)**.** *Fix $T \geq 1$ and $1 \leq W \leq T$, with the standard consecutive W-window partition. Before choosing x_t, the learner observes S_t, $\mathcal{X}_t^+$, and the exact current values $\{r_t(x), g_t(x, q_t) : x \in \mathcal{X}_t^+\}$, but not future coefficients or counterfactual states. The environment fixes its transition, reward, and pressure maps before play, while realized states may depend on executed actions. Assume $0 \leq r_t(x) \leq 1$, $\|g_t(x, q_t)\|_2 \leq 1$, and an always-available fallback satisfying $g_t(0, q_t) \preceq -\frac{1}{2}\mathbf{1}$. Let $y^{(W)}$ be a reward-maximizing W-window-safe realized-state comparator and define*

$$\mathfrak{M}_{T,W} = \inf_A \sup_\omega \mathbb{E}_\omega \left[\left[\mathrm{Reg}_T(y^{(W)})\right]_+ + \tfrac{1}{2}\mathrm{Viol}_T \right].$$

The expectation is over learner randomization; the supremum includes oblivious instances.

Definition 3 strengthens post-hoc L1 by moving validated current candidate values to decision time, thereby isolating future allocation uncertainty from value estimation. Section 4 returns to deployment, where CVQ acts on surrogates and post-hoc L1 validation measures coefficient mismatch through δ_t and search suboptimality through ξ_t.

Theorem 2 (Full-information window lower bound)**.** *For universal $c > 0$, every even $W \geq 2$ dividing T satisfies*

$$\mathfrak{M}_{T,W} \geq c\sqrt{WT}.$$

The lower bound already holds on a fixed one-state two-action routing subclass with scalar pressure, exact current rewards/pressures revealed before every action, and an oblivious reward sequence. For $W = 1$, $\mathfrak{M}_{T,1} = 0$.

The lower bound is a within-window allocation problem: the comparator spends its safe allowance in the better half, while a stopped random walk hides that half from the learner. Yao's principle yields the $\sqrt{WT}$ scaling (Appendix L). Section 4 asks whether an executable controller can match this order while exposing approximation error.

4 CVQ-Routing: From the Benchmark to a Certificate

4.1 A Realized-State Controller and Its Approximation Errors

Fix any realized-state comparator sequence $y_t \in \mathcal{X}_t^+$ for evaluation. CVQ itself does not use y when acting: it maintains $\lambda_t \in \mathbb{R}_+^K$, receives decision-time surrogates $(\hat{r}_t, \hat{g}_t)$, and approximately maximizes

$$\widehat{L}_t(x) = \hat{r}_t(x) - \lambda_t^\top \hat{g}_t(x, q_t),$$

over $\mathcal{X}_t^+$. Validated executed pressure updates the queue (Neely, 2010; Liu et al., 2023; Stradi et al., 2024). The executed value $g_t(x_t, q_t)$ used in the update must be available before round $t+1$; only values for unexecuted candidates may arrive post hoc for L1 audit:

$$\lambda_{t+1} = [\lambda_t + \eta g_t(x_t, q_t)]_+. \tag{5}$$

For additive path rewards and equation 2, the reduced edge length is $\ell_{e,t} = -\hat{\theta}_{e,t} + (\widehat{M}_t(q_t)^\top \lambda_t)_e$. The experimental reserve coordinate uses exposure $\nu_{e,t}/\max\{q_{e,t} - \rho_e, \varepsilon\}$, where ρ_e is the reserve floor and $\varepsilon > 0$ stabilizes the denominator. Thus scarce edges receive a higher pressure price before they become infeasible.

An audit must distinguish a bad surrogate from an incomplete search. With $L_t(x) = r_t(x) - \lambda_t^\top g_t(x, q_t)$, define

$$\delta_t = \sup_{x \in \mathcal{X}_t^+} |L_t(x) - \widehat{L}_t(x)|, \quad \xi_t = \sup_{x \in \mathcal{X}_t^+} \widehat{L}_t(x) - \widehat{L}_t(x_t), \quad \zeta_t = 2\delta_t + \xi_t. \tag{6}$$

Here δ_t measures decision-time versus validated L1 mismatch, whereas ξ_t measures search loss against the *audited* candidate set. These quantities are observable only when that reference set and its values are retained.

4.2 One Certificate, Then Comparator-Specific Debt

Theorem 3 (General realized-state certificate)**.** *Assume $\lambda_1 = 0$. For every realized path, every $\eta > 0$, and every L1 comparator sequence $y_t \in \mathcal{X}_t^+$,*

$$\mathrm{Reg}_T(y) + \frac{\|\lambda_{T+1}\|_2^2}{2\eta} \le \frac{\eta}{2} \sum_{t=1}^T \|g_t(x_t, q_t)\|_2^2 + \sum_{t=1}^T \zeta_t + \mathcal{D}_T(y). \tag{7}$$

Moreover $\mathrm{Viol}_T \le \|\lambda_{T+1}\|_2/\eta$.

Approximate optimality and queue drift telescope into this certificate (Appendix J). Only $\mathcal{D}_T(y)$ depends on the comparator; window safety controls that term.

Theorem 4 (W-window-safe comparator guarantee)**.** *Under Theorem 3, assume additionally that $\|g_t(x, q_t)\|_2 \le B_g$ on realized feasible actions, and let y be W-window-safe with block lengths $L_j = |B_j|$. Then*

$$\mathcal{D}_T(y) \le \frac{\eta B_g^2}{2} \sum_{j=1}^J L_j(L_j - 1) \le \frac{\eta B_g^2}{2}(W-1)T, \tag{8}$$

and therefore

$$\mathrm{Reg}_T(y) + \frac{\|\lambda_{T+1}\|_2^2}{2\eta} \le \frac{\eta B_g^2 W T}{2} + \sum_{t=1}^T \zeta_t. \tag{9}$$

When $\delta_t = \xi_t = 0$ for all t and $\eta = T^{-1/2}$, the regret is $O(W\sqrt{T})$; it is $O(\sqrt{T})$ for fixed audit windows and remains sublinear whenever $W = o(\sqrt{T})$. Under the additional bounded-reward and strict-fallback assumptions of Corollary 1, the same stepsize also gives $O(\sqrt{T})$ learner violation.

Figure 3 summarizes the two analytic contributions to the certificate. Window pressure sums nonpositively, but variation in queue prices creates the weighted debt. Proposition 5 in Appendix K constructs a bounded scalar instance with $\mathcal{D}_T(y) = \eta B_g^2 WT/4$, showing that the W dependence is order-tight up to constants under bounded pressure and window safety. Realized debt can be smaller and may have either sign.

For $W = 1$, $\mathcal{D}_T(x^\star) \le 0$ yields the pointwise certificate (Corollary 4). Weak fallback gives $O(\sqrt{T})$ regret and $O(T^{3/4})$ violation when $\delta_t = \xi_t = 0$ for all t and $\eta = T^{-1/2}$ (Corollary 5). A strict fallback margin yields the sharper queue control used below.

Corollary 1 (Strict safe-fallback margin)**.** *If $0 \le r_t(x) \le B_r$, $\|g_t(x, q_t)\|_2 \le B_g$ on realized feasible actions, $g_t(0, q_t) \preceq -\sigma \mathbf{1}$, and $\delta_t = \xi_t = 0$ for all t, then $\|\lambda_t\|_2 \le B_r/\sigma + \eta B_g^2/(2\sigma) + \eta B_g$ and $\mathrm{Viol}_T = O(1/\eta)$. Thus $\eta = T^{-1/2}$ yields $O(\sqrt{T})$ pointwise-safe regret and $O(\sqrt{T})$ violation.*

Since $g_t(0, q_t) = -b_t$, the strict-margin condition strengthens the weak fallback in equation 2 to uniform slack $b_t \succeq \sigma \mathbf{1}$. It is used only for the $O(1/\eta)$ queue bound and the matching regret–violation frontier, not for Theorems 3 and 4.

4.3 The Attainable Frontier and Executable Audit Loop

Corollary 2 (Window-aware regret–violation frontier)**.** *Under Theorem 4 and Corollary 1, for any W-window-safe y with $\delta_t = \xi_t = 0$ for all t,*

$$\mathrm{Reg}_T(y) \le \frac{\eta B_g^2 W T}{2}, \quad \mathrm{Viol}_T \le \frac{B_r}{\sigma\eta} + C_\sigma, \qquad C_\sigma = \frac{B_g^2}{2\sigma} + B_g. \tag{10}$$

For a weight $\gamma > 0$, the proved envelope of $\mathrm{Reg}_T(y) + \gamma \mathrm{Viol}_T$ is minimized at $\eta_\gamma^* = \sqrt{2\gamma B_r/(\sigma B_g^2 W T)}$, giving at most $B_g\sqrt{2\gamma B_r W T/\sigma} + \gamma C_\sigma$. Thus $\eta = \Theta((WT)^{-1/2})$ gives $O(\sqrt{WT})$ regret and violation simultaneously, whereas $\eta = \Theta(T^{-1/2})$ gives $O(W\sqrt{T})$ regret and $O(\sqrt{T})$ violation.

This choice minimizes the *proved envelope*; realized deployment reward is assessed separately in the experiments. Combined with the full-information lower bound, it matches the minimax order in the normalized game.

Corollary 3 (Minimax phase transition for exact L1)**.** *Under the normalized exact-L1 game of Definition 3, there are universal constants $0 < c < C < \infty$ such that $\mathfrak{M}_{T,1} = 0$ for every $T \ge 1$ and, for every integer $2 \le W \le T$,*

$$c\sqrt{WT} \le \mathfrak{M}_{T,W} \le C\sqrt{WT}.$$

For the upper bound, specialize Corollary 2 to the normalized game with $B_r = B_g = 1$, $\sigma = 1/2$, and weight $\gamma = 1/2$. It gives a deterministic nonnegative bound on $\mathrm{Reg}_T(y^{(W)})$, hence the same bound on $[\mathrm{Reg}_T(y^{(W)})]_+$; choosing $\eta = \Theta((WT)^{-1/2})$ yields $O(\sqrt{WT})$. Theorem 2 gives the even divisible lower bound, and Appendix L extends it to arbitrary W, T by neutral padding without changing the order.

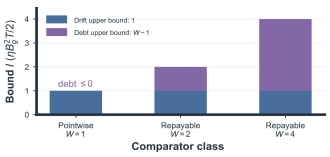

Figure 3: **Repayment enlarges the comparator-debt envelope at fixed audit level.** After normalization by $\eta B_g^2 T/2$, drift contributes at most 1 and window debt at most $W - 1$. Bars show the analytic terms for zero approximation charge; empirical debt and minimax cost are reported separately.

The $W = 1$ endpoint is distinct: repayment introduces allocation uncertainty even with exact current feedback. Corollaries 2–3 characterize the exact-information frontier and the operating point obtained by minimizing the proved certificate envelope. Deployment follows Algorithm 1, acting on $(\hat{r}_t, \hat{g}_t)$ and using post-hoc L1 values to audit coefficient mismatch and search suboptimality.

Algorithm 1 CVQ-Routing with L1 audit logging

1: Initialize q_1, $\lambda_1 = 0$, and step size η.
2: **for** $t = 1, \ldots, T$ **do**
3: Observe S_t, $\mathcal{X}_t^+$, and surrogates $(\hat{r}_t, \hat{g}_t)$.
4: Choose x_t such that $\widehat{L}_t(x_t) \ge \sup_{x \in \mathcal{X}_t^+} \widehat{L}_t(x) - \xi_t$.
5: Execute x_t; obtain the validated executed pressure, update q_t by equation 1, and update λ_t by equation 5.
6: Retain the candidate set, surrogates, action, queue, and validated candidate values for post-hoc L1 audit.
7: **end for**

With an explicit catalog, route scoring reduces to finite enumeration. On a DAG, additive reduced lengths can be optimized by topological dynamic programming even when some edge lengths are negative. For cyclic graphs, exact shortest-path computation requires no relevant reachable negative-cost cycle. Approximate search contributes to ξ_t without changing the audit level, provided the larger reference catalog remains available post hoc (Appendices P and Q).

5 Experiments

The experiments are organized around four mechanisms in the theory. The exact W-window sweep examines how comparator strength changes with the repayment horizon in Theorem 4 and Corollary 3.

Forecast-noise and route-search-budget sweeps isolate the δ_t and ξ_t approximation charges in Theorem 3. The stepsize sweep traces the regret–violation operating points in Corollary 2. Finally, the public reset replay examines the L1–L2 state-alignment distinction in Definition 1 and Proposition 3.

• **Controlled synthetic tests.:** Layered DAGs expose the quantities entering the certificate while keeping the realized candidate set and resource dynamics explicit. The exact-window experiment solves the W-window comparator on the same audited L1 trajectories. Forecast-noise and limited-search sweeps vary the two approximation channels separately, while the stepsize sweep varies the controller operating point. We additionally report finite-horizon reward–violation profiles over $T \in \{200, 400, 800, 1600\}$. CVQ-Oracle provides a simulation-only exact-information reference, while NoReserve and NoReject isolate the reserve-price and fallback mechanisms. Complete seven-method operating points are reported in Appendix R.3.

• **Public reset test.:** The Abilene/QKD replay compares same-state and own-trajectory evaluation by first scoring each target action on the CVQ-realized state sequence and then resetting the simulator and executing the target rule on its own trajectory. The replay uses the 12-node, 30-edge Abilene topology, 12 chronological report weeks, and measured QKD secure-key-rate calibration. Public baselines additionally include ShortestAvailable-QKD and WidestShortest-QKD.

Table 1: **Experimental settings.**

Quantity	Setting
Synthetic horizons	$T \in \{200, 400, 800, 1600\}$
Report seeds	12 paired seeds
Validation	seeds 100–103, $T = 800$
Repayment windows	$W \in \{1, 2, 4, 8, 16, 32\}$
Exact-window test	$T = 400$, eight-route catalog
Public topology	Abilene, 12 nodes / 30 directed edges
Report weeks	12 chronological weeks
Normalization	all 24 weeks
QKD input	measured SKR calibration
Intervals	$1.96s/\sqrt{n}$

• **Selection and reporting.:** Parameterized controllers use equally sized validation grids. Hyperparameters are selected on seeds 100–103 at $T = 800$ and then frozen for the 12 report seeds. Unless otherwise specified, intervals are $1.96s/\sqrt{n}$; for the 12 chronological public-replay weeks, they are descriptive rather than independence-based. Appendices R and T provide the full grids and extended results.

5.1 Comparator Strength and the L1–L2 Difference

At $T = 400$, exact window comparators are solved on the same eight-route L1 catalog. Increasing W from 1 to 32 raises comparator reward from $.109 \pm .008$ to $.597 \pm .005$ and the positive-pressure action rate from 0 to $.551 \pm .007$ (Figure 4). All 72 seed/window certificates pass, with comparator debt remaining negative. Thus, on this nested powers-of-two grid, a longer repayment horizon enlarges the realized-state comparator class without changing the L1 record.

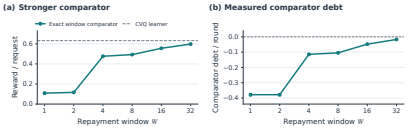

Figure 4: **Longer repayment windows strengthen the comparator without changing the audit record.** Exact comparator reward and measured debt on the same realized L1 trajectories ($T = 400$, 12 seeds; normal 95% intervals). The dashed reward reference is the learner; measured debt remains negative at every tested W.

The public reset experiment instead changes the information interface. L1 evaluates each target rule on the CVQ-realized states, whereas L2 resets the simulator and executes that rule on its own trajectory. Under scarcity, the L1–L2 gaps rise to $.334 \pm .017$ for RewardOnly, $.319 \pm .016$ for Shortest, and $.318 \pm .016$ for Widest, while SafeGreedy yields $-.160 \pm .003$ (Figure 5). The sign reversal shows that the discrepancy is policy- and trajectory-dependent rather than a single policy-independent correction.

5.2 Approximation Errors and Controller Mechanisms

The scarcity ablation separately probes controller mechanisms. Removing the reserve price or fallback substantially increases normalized violation. Full ablation values are reported in Appendix R.6, and horizon-wise baseline operating points in Appendix R.3. The stepsize sweep likewise traces the certificate tradeoff in Corollary 2; and horizon-wise baseline operating points in Appendix R.3. The

stepsize sweep likewise traces the operating points predicted by Corollary 2; its purpose is to examine the certified regret–violation envelope rather than select a single universal stepsize.

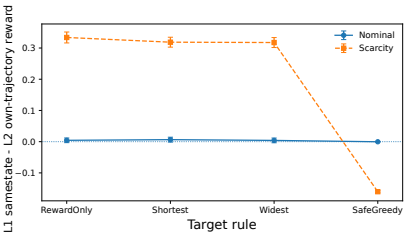

Figure 5: **Public Abilene/QKD replay separates same-state L1 from own-trajectory L2 evaluation under scarcity.** Points show $\bar{V}_{\mathrm{L1}} - \bar{V}_{\mathrm{L2}}$ for four target rules. In the nominal regime the gaps remain near zero. Under the predeclared scarcity stress, RewardOnly, Shortest, and Widest have large positive gaps, whereas SafeGreedy changes sign. Intervals are normal 95% intervals over 12 chronological report weeks and are descriptive of week-to-week variation.

Table 2: **Comparator-strength and state-alignment diagnostics.** Panel A varies the repayment window; Panel B reports the public Abilene/QKD L1–L2 gap under nominal and scarcity regimes.

Panel A: Controlled comparator mechanism

W	Comp. reward	Pos. pressure (%)	Debt / round
1	$.109 \pm .008$	0.0 ± 0.0	$-.379 \pm .007$
2	$.116 \pm .007$	1.0 ± 0.2	$-.379 \pm .007$
4	$.476 \pm .006$	41.6 ± 0.7	$-.114 \pm .004$
8	$.493 \pm .005$	43.3 ± 0.6	$-.105 \pm .005$
16	$.555 \pm .004$	50.4 ± 0.6	$-.048 \pm .003$
32	$.597 \pm .005$	55.1 ± 0.7	$-.017 \pm .005$

Panel B: Public Abilene/QKD state-alignment gap

Target rule	Nominal gap	Scarcity gap
RewardOnly	$.004 \pm .007$	$.334 \pm .017$
ShortestAvailable-QKD	$.007 \pm .008$	$.319 \pm .016$
WidestShortest-QKD	$.004 \pm .008$	$.318 \pm .016$
SafeGreedy	$-.000 \pm .001$	$-.160 \pm .003$

Panel A uses fixed L1 trajectories across W; Panel B reports the L1–L2 reward gap. Values are mean $\pm$ normal 95% intervals over 12 seeds/weeks.

5.3 Robustness, Compute, and Boundary Regimes

The robustness suite varies bursts, trust-edge removal, replenishment, and graph scale. At burst probability .30, CVQ/AdaGrad achieve rewards $.724/.748$ with normalized violations $.173/.151$. In public replay, random measured-SKR perturbations leave CVQ reward and aggregate violation nearly unchanged, while minimum reserve falls from 21.72 to -15.57, separating cumulative pressure from a hard reserve-floor criterion.

Forecast noise and restricted search increase the approximation charge: $\sum_t \zeta_t$ rises from 0 to 141.2 across the noise sweep and reaches 371.0 when only four of 64 audited routes are inspected. All 192 sensitivity runs satisfy the pathwise certificate (Figure 6).

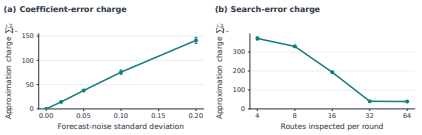

Figure 6: **Two auditable approximation channels.** (a) Forecast-noise sweep: measured coefficient-mismatch charge. (b) Search-budget sweep: measured search charge against the same 64-route audited reference set. Means and normal 95% intervals over 12 seeds.

At 32 routes, CVQ/AdaGrad/EG require $.0214/.0203/.0204$ ms per round; CVQ peak traced memory is about .030 MB. Increasing the catalog from 8 to 64 routes changes CVQ runtime from .0163 to .0200 ms and memory from .0170 to .0390 MB. Full stress, trajectory, and compute records are in Appendices R.8, U, and V.

6 Discussion and Conclusion

A deployment record determines which alternative decisions can be audited, while the comparator safety class determines the cost of supporting those comparisons. The L0/L1/L2 hierarchy encodes the information axis and the repayment window W the comparator axis; in the normalized exact-L1 game, their interaction yields the $\Theta(\sqrt{WT})$ frontier. This separation gives a simple rule for deployment claims: state the audit level that supports the comparison and the safety class of the comparator. Exact decision-time L1 isolates allocation uncertainty, post-hoc L1 supports realized-state alternatives, and L2 adds counterfactual-state reconstruction. CVQ-Routing is one executable realization of this framework, with a pathwise certificate that separates coefficient mismatch, search loss, queue drift, and comparator debt. The experiments probe the corresponding repayment, approximation, and state-alignment regimes. More broadly, an online performance claim is only as meaningful as the audit interface that identifies it and the comparator class that it asks the learner to match.

AI USE STATEMENT

Generative AI tools were used for language editing, structural refinement, formatting assistance, and exploratory discussion during manuscript preparation. The authors are responsible for all theoretical claims, experimental results, analyses, and final conclusions.

ETHICS STATEMENT

This work does not involve human-subject experimentation; the empirical study uses public network measurement data and controlled simulation.

REPRODUCIBILITY STATEMENT

The supplement contains the experiment code, processed traces, result outputs, and deterministic panel-by-panel figure scripts. Formal statements and proofs are indexed in Appendix E; full protocols, baselines, seeds, and measurements are in Appendices R, T, and V. Figure reproduction uses the bundled outputs without network access. Rerunning public experiments additionally requires the four QKD measurement CSVs; a checksum-verified downloader and separate clean-output runner are included. The supplement distinguishes output checks, numerical reruns, and hardware-dependent timings.

REFERENCES

Oren Anava, Elad Hazan, and Shie Mannor. Online learning for adversaries with memory: Price of past mistakes. In *Advances in Neural Information Processing Systems*, volume 28, pp. 784–792, 2015.

Raman Arora, Ofer Dekel, and Ambuj Tewari. Online bandit learning against an adaptive adversary: from regret to policy regret. In *Proceedings of the 29th International Conference on Machine Learning*, pp. 1503–1510. Omnipress, 2012.

Baruch Awerbuch and Robert D. Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, pp. 45–53. Association for Computing Machinery, 2004.

Martino Bernasconi, Matteo Castiglioni, Andrea Celli, and Federico Fusco. Beyond primal-dual methods in bandits with stochastic and adversarial constraints. In *Advances in Neural Information Processing Systems*, volume 37, pp. 8541–8568, 2024a.

Martino Bernasconi, Matteo Castiglioni, Andrea Celli, and Federico Fusco. Bandits with replenishable knapsacks: the best of both worlds. In *The Twelfth International Conference on Learning Representations*, 2024b.

Martino Bernasconi, Matteo Castiglioni, and Andrea Celli. No-regret is not enough! Bandits with general constraints through adaptive regret minimization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 3877–3898. PMLR, 2025.

Kush Bhatia and Karthik Sridharan. Online learning with dynamics: A minimax perspective. In *Advances in Neural Information Processing Systems*, volume 33, pp. 15020–15030, 2020.

Mark Braverman, Jingyi Liu, Jieming Mao, Jon Schneider, and Eric Xue. A new benchmark for online learning with budget-balancing constraints. arXiv preprint arXiv:2503.14796, 2025.

Matteo Castiglioni, Andrea Celli, and Christian Kroer. Online learning under budget and ROI constraints via weak adaptivity. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5792–5816. PMLR, 2024.

Nicolò Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

Leandro C. Coelho, Jean-François Cordeau, and Gilbert Laporte. Thirty years of inventory routing. *Transportation Science*, 48(1):1–19, 2014.

Bernard Fortz and Mikkel Thorup. Internet traffic engineering by optimizing OSPF weights. In *Proceedings IEEE INFOCOM 2000*, volume 2, pp. 519–528, 2000.

Eric C. Hall and Rebecca Willett. Dynamical models and tracking regret in online convex programming. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 579–587. PMLR, 2013.

Yuanchen Hao, Yuheng Xie, Wenpeng Gao, and Jianjun Tang. Q-learning for resource-aware and adaptive routing in trusted-relay QKD network. *Photonics*, 12(10):969, 2025. doi: 10.3390/photonics12100969.

Samir Khan, Martin Saveski, and Johan Ugander. Off-policy evaluation beyond overlap: Sharp partial identification under smoothness. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 23734–23757. PMLR, 2024.

Pavel Kolev, Georg Martius, and Michael Muehlebach. Online learning under adversarial nonlinear constraints. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53227–53238, 2023.

Raunak Kumar, Sarah Dean, and Robert D. Kleinberg. Online convex optimization with unbounded memory. In *Advances in Neural Information Processing Systems*, volume 36, pp. 26229–26270, 2023.

Nikolaos Liakopoulos, Apostolos Destounis, Georgios Paschos, Thrasyvoulos Spyropoulos, and Panayotis Mertikopoulos. Cautious regret minimization: Online optimization with long-term budget constraints. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3944–3952. PMLR, 2019.

Mingyang Liu, Asuman Ozdaglar, Tiancheng Yu, and Kaiqing Zhang. Regret minimization with adaptive opponents in repeated games. In *Proceedings of the 39th Conference on Learning Theory*, volume 336 of *Proceedings of Machine Learning Research*, pp. 4857–4858. PMLR, 2026.

Xin Liu, Zixian Yang, and Lei Ying. Online nonstochastic control with adversarial and static constraints. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22277–22288. PMLR, 2023.

Jorge López, Charalampos Chatzinakis, and Marc Cartigny. Route planning and online routing for quantum key distribution networks. In *2026 IEEE 12th International Conference on Network Softwarization (NetSoft)*, pp. 267–272, 2026.

Wanteng Ma, Dong Xia, and Jiashuo Jiang. High-dimensional linear bandits with knapsacks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34008–34037. PMLR, 2024.

Ahmed M. Medhat, Tarik Taleb, Asma Elmangoush, Giuseppe A. Carella, Stefan Covaci, and Thomas Magedanz. Service function chaining in next generation networks: State of the art and research challenges. *IEEE Communications Magazine*, 55(2):216–223, 2017.

Michael J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Synthesis Lectures on Communication Networks. Morgan & Claypool Publishers, 2010.

Momtchil Peev, Christoph Pacher, Romain Alléaume, Claudio Barreiro, Jan Bouda, Winfried Boxleitner, Thierry Debuisschert, Eleni Diamanti, Mehrdad Dianati, James F. Dynes, et al. The SECOQC quantum key distribution network in Vienna. *New Journal of Physics*, 11(7):075001, 2009. doi: 10.1088/1367-2630/11/7/075001.

M. Sasaki, M. Fujiwara, H. Ishizuka, W. Klaus, K. Wakui, M. Takeoka, S. Miki, T. Yamashita, Z. Wang, A. Tanaka, K. Yoshino, Y. Nambu, S. Takahashi, A. Tajima, A. Tomita, T. Domeki, T. Hasegawa, Y. Sakai, H. Kobayashi, T. Asai, K. Shimizu, T. Tokura, T. Tsurumaru, M. Matsui, T. Honjo, K. Tamaki, H. Takesue, Y. Tokura, J. F. Dynes, A. R. Dixon, A. W. Sharpe, Z. L. Yuan, A. J. Shields, S. Uchikoga, M. Legré, S. Robyr, P. Trinkler, L. Monat, J.-B. Page, G. Ribordy, A. Poppe, A. Allacher, O. Maurhart, T. Länger, M. Peev, and A. Zeilinger. Field test of quantum key distribution in the Tokyo QKD network. *Optics Express*, 19(11):10387–10409, 2011.

Francesco Emanuele Stradi, Jacopo Germano, Gianmarco Genalti, Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Online learning in CMDPs: Handling stochastic and adversarial constraints. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 46692–46721. PMLR, 2024.

Masatoshi Uehara, Chengchun Shi, and Nathan Kallus. A review of off-policy evaluation in reinforcement learning. *Statistical Science*, 41(3):561–581, 2026.

Appendix

CONTENTS

A Notation **15**

B Extended related work and representative-work comparison **15**

C Extended setup, information model, and benchmark hierarchy **16**

D Assumptions, operational meaning, and consequences **17**

E Proof map and technical lemmas **18**

F L0 non-identification proof **18**

G Safety–regret tradeoff for unrestricted L1 comparators **18**

H Counterfactual policy lower bound **18**

I Proof of the state-alignment decomposition **19**

J Proof of the generalized realized-state certificate **19**

K Window-safe comparator proof **19**

L Minimax lower bound for the exact-L1 window game **20**

M Weak-safe violation bound **22**

N Strict-margin violation bound **23**

O Window-aware rate frontier proof **23**

P Algorithm and route computation **23**

Q Computational regimes and oracle error **24**

R Experimental evidence matrix and complete protocol **24**

R.1 Synthetic environment and paired randomness 24

R.2 Equal-budget baseline tuning 25

R.3 Full $T = 800$ Endpoint Comparison 26

R.4 Full $T = 800$ endpoint table 26

R.5 Window-safe comparator strength 26

R.6 Targeted scarcity ablation 27

R.7 Stepsize and error-channel sensitivity 29
R.8 Robustness axes 30
R.9 Efficiency and implementation cost 33

S Certificate validity and error-channel diagnostics **33**

T Public-data controlled replay and QKD-native baselines **33**

U Boundary regimes and qualitative interpretation **36**

V Reproducibility, compute, code map, and resources **37**

APPENDIX ROADMAP

Appendix A collects recurring notation; Appendix B gives the extended related-work comparison; Appendices C and D expand the information model and theorem conditions; Appendices E, F, G, H, I, J, K, L, M, N, and O contain the proof tree; Appendices P and Q specify route computation; and Appendices R, R.5, S, T, U, and V give the full experimental and reproducibility record.

A NOTATION

Table 3 collects the symbols that recur across the information model, comparator definitions, certificates, and experiments. Central objects are defined at first use in the main text; Table 3 collects them for reference.

Table 3: Core notation used across the audit hierarchy, comparator family, and certificates.

Symbol	Meaning	Role / audit level
$G = (V, E)$	directed edge universe	environment
q_t	pre-decision edge inventory vector	realized state
G_t^{op}	operational graph after availability, trust, and inventory filters	L0/L1 state
$\mathcal{X}_t^+$	realized feasible route catalog plus fallback 0	L1 candidate set
x_t	learner action at round t	executed action
S_t	realized pre-action state $(q_t, G_t^{\mathrm{op}}, \mathcal{X}_t^+)$	L0/L1
$r_t(x)$	validated same-state reward of candidate x	L1
$g_t(x, q_t)$	K-dimensional long-run pressure vector	L1 / safety
$M_t(q_t), b_t$	additive pressure coefficients in $g_t = M_t(q_t)x - b_t$	L1 telemetry
$\hat{r}_t, \hat{g}_t$	deployment-time surrogate reward/pressure	controller input
λ_t	nonnegative virtual queue / shadow price	controller state
$y_{1:T}$	comparator sequence on learner-realized states	L1 benchmark
$y^{(W)}$	reward-maximizing W-window-safe comparator in the exact-L1 game	minimax benchmark
$\mathrm{Reg}_T(y)$	same-state cumulative regret to comparator y	L1 target
PolReg_T	policy regret evaluated on a comparator's own state trajectory	L2 target
Viol_T	positive part of cumulative learner pressure	safety metric
$\mathcal{D}_T(y)$	queue-weighted comparator pressure $\sum_t \lambda_t^\top g_t(y_t, q_t)$	auditable debt
δ_t	surrogate-vs-validated Lagrangian coefficient error	forecast error
ξ_t	route-search/catalog suboptimality	oracle error
$\zeta_t = 2\delta_t + \xi_t$	total per-round approximation charge	certificate error
W	comparator repayment-window length	comparator strength
η	virtual-queue stepsize	reward–safety operating point
B_r, B_g, σ	reward bound, pressure bound, strict fallback margin	theorem constants

B EXTENDED RELATED WORK AND REPRESENTATIVE-WORK COMPARISON

The main text gives the conceptual comparison; this section makes the information and comparator axes explicit. The matrix in Table 4 summarizes the representative formulations cited here. A ✓ means the object is explicit in that formulation, ○ means it is available with additional support/model assumptions or in formulation-dependent variants, and "–" means it lies outside the primary target. The cells describe interface scope rather than method merit, and each row refers to the cited representative formulation rather than an entire literature family.

Constrained online learning and long-run constraints. Virtual queues, online primal–dual updates, and constrained online-learning methods provide the controller machinery used by CVQ (Neely, 2010; Liu et al., 2023; Kolev et al., 2023; Stradi et al., 2024). Work on weak adaptivity and adversarial constraints studies when stronger online guarantees remain possible (Castiglioni et al., 2024; Bernasconi et al., 2024a; 2025). Our distinction is orthogonal to the update rule: before asking which controller attains low regret, we ask which comparator value is auditable from the record that controller leaves behind.

Window-balanced comparators. The K-benchmark of Liakopoulos et al. (2019) and pacing-over-windows benchmarks of Braverman et al. (2025) show that budget repayment windows interpolate comparator strength. We use the same structural idea after fixing the audit interface. This yields a two-axis decomposition: audit level determines which values are measurable, while W determines how much temporary pressure an L1 comparator may borrow and repay.

Policy regret and adaptive environments. Policy regret was introduced to avoid freezing an adaptive loss sequence when a different action history would have changed future outcomes (Arora et al., 2012); online learning with memory and stateful dynamics makes the same counterfactual dependence explicit (Anava et al., 2015; Bhatia & Sridharan, 2020). Repeated Policy Regret further strengthens history-dependent comparators in repeated games (Liu et al., 2026). These are natural L2 objects in our terminology: evaluating them requires a counterfactual trajectory, not merely all actions on the trajectory actually visited.

Off-policy evaluation and support. OPE asks whether the value of a target policy can be identified from logged data and highlights overlap, model structure, and partial identification when support is incomplete (Uehara et al., 2026; Khan et al., 2024). L0→L1 is analogous to an action-outcome support question on a realized state. L1→L2 is different: the missing object is the state and future feasible set induced by a different history.

Resource-aware and QKD routing. Inventory routing, traffic engineering, and resource-aware QKD routing model how scarce resources change route quality or availability (Coelho et al., 2014; Fortz & Thorup, 2000; López et al., 2026; Hao et al., 2025). Their operational objectives motivate our case study and baselines. Our additional question is evaluative: once route choices change future feasibility, a performance statement must distinguish a same-state audit from a counterfactual-trajectory comparison. Even oracle-accurate same-state values leave the alternative state trajectory as a separate object.

Table 4: Representative formulations by information and comparator axis. Symbols describe whether an object is explicit (✓), conditional/model-dependent (○), or outside the primary target (–).

Representative family	Same-state unchosen values	Alt.-trajectory states	Long-run safety	Repayment window	Explicit audit hierarchy
Constrained online learning (Neely, 2010; Liu et al., 2023)	○	–	✓	○	–
Window-budget benchmarks (Liakopoulos et al., 2019; Braverman et al., 2025)	○	–	✓	✓	–
Policy regret / dynamics (Arora et al., 2012; Bhatia & Sridharan, 2020)	○	✓	–	–	–
OPE / support (Uehara et al., 2026; Khan et al., 2024)	○	○	–	–	○
Resource-aware / QKD routing (Coelho et al., 2014; López et al., 2026; Hao et al., 2025)	○	–	○	–	–
This work	✓ (L1)	✓ (L2)	✓	✓	✓

C Extended setup, information model, and benchmark hierarchy

This section expands Definition 1 and Equation 3 and fixes exactly what is observed, audited, or counterfactual under the same information assumptions used in the main text.

Level 0: executed-action outcome log. At round t, L0 records the realized pre-action state S_t, the executed action x_t, and realized feedback for x_t. This can be sufficient to update an online controller, but it does not identify the reward or safety of every unchosen action. Candidate-set reconstruction alone is insufficient if outcome coefficients for those candidates are not known.

Proposition 1 (Edge telemetry realizes L1 for additive routing)**.** *Let A_t contain the incidence vectors of the realized candidate routes. If $r_t(x) = \theta_t^\top x$ and $g_t(x, q_t) = M_t(q_t)x - b_t$, and the audit record contains $(A_t, \theta_t, M_t(q_t), b_t)$, then every same-state candidate value is reconstructed exactly as*

$$A_t\theta_t, \qquad A_t M_t(q_t)^\top - \mathbf{1}b_t^\top.$$

No physical execution of an unchosen route is required for these additive audit quantities.

Level 1: realized-state action audit. L1 additionally records/reconstructs the declared realized candidate set $\mathcal{X}_t^+$ and supplies validated post-hoc $r_t(x), g_t(x, q_t)$ for candidate actions. For additive routing, let A_t stack candidate path-incidence vectors. Logged edge reward coefficients θ_t and pressure map $(M_t(q_t), b_t)$ reconstruct all candidate values as

$$r_t(A_t) = A_t\theta_t, \qquad g_t(A_t, q_t) = A_t M_t(q_t)^\top - \mathbf{1}b_t^\top,$$

which proves Proposition 1. The audit therefore needs edge telemetry, not physical execution of every unchosen path. If outcomes contain action-specific latent effects that cannot be reconstructed in this way, they remain an L0 support problem. Deployment-time surrogates $(\hat{r}_t, \hat{g}_t)$ need not equal the post-hoc coefficients; their discrepancy is measured by δ_t.

Level 2: counterfactual-state audit. L2 additionally reconstructs S_t^π, its feasible candidate set, and the corresponding reward/pressure values for a policy that would have chosen different earlier actions. Reset experiments, shadow deployments, or a validated dynamics-and-outcome model can provide this interface. Merely observing all actions at the learner's realized state does not provide it.

Benchmark hierarchy. Comparisons across W are nested when the partition is coarsened, as on the powers-of-two experimental grid; a larger maximum block length alone does not imply nesting under unrelated boundaries.

The L1 family in equation 3 ranges from pointwise safety to progressively longer repayment windows. $W = 1$ requires $g_t(y_t, q_t) \preceq 0$ every round. For $W > 1$, Definition 2 allows positive pressure inside a block provided the block sum is nonpositive; the class is strictly larger whenever safe and pressure-producing actions can be interleaved. Theorem 4 controls the queue-weighted debt of this stronger class by queue variation and gives an $O(W\sqrt{T})$ regret rate under exact coefficients and exact route optimization. As W grows, the benchmark approaches a coarse cumulative-safety comparison and the guarantee weakens accordingly. An unrestricted L1 comparator is still auditable, but Proposition 4 shows that no algorithm can simultaneously guarantee sublinear regret to every such comparator and sublinear safety violation. A policy comparator on S_t^π requires the separate L2 alignment interface. Thus this section verifies the main-text information contract: L1 is sufficient for the realized-state comparator family, but not for an own-trajectory policy claim.

D Assumptions, operational meaning, and consequences

This section records where each condition is used, its operational meaning, and the consequence of relaxing it.

Table 5: Assumption and information ledger.

Condition	Role	Operational meaning	Consequence / diagnostic
L1 candidate/action coefficients	Audit	Same-state unchosen actions can be evaluated post hoc	Without them, Proposition 2 applies; exact same-state regret is not identified.
Logged realized candidate set	Audit	The action set actually available at S_t is reconstructable	Missing candidates make ξ_t unverifiable; route-catalog sweeps quantify known truncation.
Weak-safe fallback	Benchmark nonemptiness / weak rate	$0 \in \mathcal{X}_t^+$ and $g_t(0, q_t) \preceq 0$	Ensured by equation 2; it is not required by Theorem 3, and it supports the pointwise/weak-safe specializations.
Window safety W	Comparator strength	Comparator pressure is repaid within consecutive blocks of length at most W	Theorem 4 and Corollary 2 give the two rate regimes; exact comparators are solved for $W = 1$–32.
Bounded reward/pressure	Rate constants	The general certificate retains the realized $\sum_t \|g_t(x_t, q_t)\|_2^2$ and needs no uniform pressure bound; the window, weak/strict, and minimax specializations impose uniform bounds	Restoring physical units rescales constants; identification is unchanged.
Exact-L1 minimax normalization	Game-level rate	$0 \le r \le 1$, $\|g\|_2 \le 1$, fallback margin $1/2$, and the standard consecutive W-partition	These are exactly Definition 3; Theorem 2 and Corollary 3 use no stronger game assumptions.
Strict margin $\sigma > 0$	Sharper rate only	Fallback creates uniformly negative pressure	If absent, Corollary 5 remains; the released experiments do not isolate a strict-margin sweep.
Zero approximation charge	Rate specialization	$\delta_t = \xi_t = 0$ for all t	Theorem 3 retains $\sum_t \zeta_t$ when surrogate or search error is present.
L2 state alignment	Policy-level claim only	Unexecuted policy states can be reconstructed	Policy-level comparison then requires an L2 alignment interface.
Route computation regime	Computation	Catalog, DAG, or no-relevant-negative-cycle shortest path	Truncation/approximation is charged by ξ_t; unrestricted negative-cycle simple-path exactness lies outside this computation regime.

The audit level determines the information object. Strict slack and boundedness affect rates, while route tractability affects computation; none changes the underlying realized-state comparison object.

E PROOF MAP AND TECHNICAL LEMMAS

The certificate branch starts from one-step projection drift and yields Theorem 3; blockwise summation by parts then gives Theorem 4. The weak/strict fallback bounds control violation, and optimizing their common stepsize with the window theorem yields Corollary 2. Specializing that frontier to the normalized exact-L1 game gives the upper half of Corollary 3, while the stopped-random-walk construction in Appendix L gives its lower half and the $W = 1$ zero-cost endpoint. Independently, Proposition 2 establishes the L0 outcome-support boundary, Proposition 4 establishes the comparator-strength boundary inside L1, while Theorem 1 gives an online policy-regret obstruction and Proposition 3 states the additional alignment condition for a policy-value comparison.

Lemma 1 (One-step queue drift)**.** *For equation 5,*

$$\frac{\|\lambda_{t+1}\|_2^2 - \|\lambda_t\|_2^2}{2\eta} \le \lambda_t^\top g_t(x_t, q_t) + \frac{\eta}{2}\|g_t(x_t, q_t)\|_2^2.$$

Proof. Projection onto $\mathbb{R}_+^K$ is nonexpansive, so $\|[\lambda_t + \eta g_t]_+\|_2^2 \le \|\lambda_t + \eta g_t\|_2^2$. Expanding and dividing by 2η gives the result. □

F L0 NON-IDENTIFICATION PROOF

Restatement and proof of Proposition 2. Consider $T = 1$, one realized state, two feasible safe actions a, b, and a logging/controller rule that executes a. In environment P, let $r(a) = 0, r(b) = 0$; in environment Q, let $r(a) = 0, r(b) = 1$. The L0 record—state, candidate labels if desired, executed action a, and executed reward 0—is identical, but regret to the best feasible action is 0 under P and 1 under Q. Therefore no estimator that is only a function of the L0 record can equal the exact same-state best-action regret in both environments. □

The construction isolates the same-state outcome-support requirement. All direct certificate experiments use L1 records containing the realized candidate coefficients needed for the corresponding comparator values.

G SAFETY–REGRET TRADEOFF FOR UNRESTRICTED L1 COMPARATORS

Proof of Proposition 4. Consider one repeated state with two fully audited actions. Action s has reward 0 and scalar pressure $g(s) = -1$; action a has reward 1 and pressure $g(a) = +1$. Let the L1 comparator choose a on all T rounds. If the learner chooses a on N rounds, then

$$\mathrm{Reg}_T(y) = T - N, \qquad \mathrm{Viol}_T = [N - (T - N)]_+ = [2N - T]_+.$$

If $N \le T/2$, then $\mathrm{Reg}_T(y) \ge T/2$. If $N > T/2$, then

$$\mathrm{Reg}_T(y) + \tfrac{1}{2}\mathrm{Viol}_T = (T - N) + \tfrac{1}{2}(2N - T) = T/2.$$

Thus every action sequence satisfies the claimed inequality. □

The example is fully L1-audited: both action rewards and pressures are known at every round. The obstruction is comparator admissibility rather than missing feedback. Restricting how long positive comparator pressure can remain unpaid, as in Definition 2, creates an intermediate class that can be strictly larger than pointwise safety while avoiding the unrestricted tradeoff.

H COUNTERFACTUAL POLICY LOWER BOUND

Restatement and proof of Theorem 1. Let $T = 2n$. There are two parallel unit-consumption edges e_1, e_2, each with initial inventory n and no replenishment. During the first n rounds both edges yield reward 1. A hidden index $h \in \{1, 2\}$, unobserved in the first half, specifies the only edge that can serve the unit-reward request in each of the last n rounds.

Fix a deterministic online algorithm. Let N_i be the number of first-half uses of e_i and $S = N_1 + N_2 \le n$. Conditional on h, at most $n - N_h$ units of e_h remain for the second half, so the learner's value is at most $S + n - N_h$. Choose h uniformly. Since $\mathbb{E}_h N_h = S/2$,

$$\mathbb{E}_h V_T(A; \omega_h) \le S + n - S/2 \le 3n/2.$$

The hindsight policy spends only the nonhidden edge in the first half and preserves all n units of e_h for the second half, obtaining $2n$. Expected regret is at least $n/2 = T/4$. Yao's minimax principle gives the randomized lower bound. □

The construction isolates online uncertainty about off-trajectory resource preservation and proves the regret lower bound in Theorem 1. Retrospective identification is a separate question: a validated model or reset protocol can supply L2 reconstruction for evaluation, while the online learner still acts without future requests.

I Proof of the state-alignment decomposition

Proof of Proposition 3. Add and subtract the aligned realized-state reward:

$$\begin{aligned} V_T(\pi) - V_T(A) &= \sum_t \big(r_t^\pi(x_t^\pi) - r_t(\tilde{x}_t^\pi)\big) + \sum_t \big(r_t(\tilde{x}_t^\pi) - r_t(x_t)\big) \\ &\le \sum_t \epsilon_t^\pi + \sum_t \big(r_t(x_t^\star) - r_t(x_t)\big), \end{aligned}$$

where pointwise safety of $\tilde{x}_t^\pi$ and optimality of $x_t^\star$ on S_t give the last step. □ If alignment can produce an infeasible/unsafe action, an additional projection/feasibility term is required. Omitting it would overstate what L1 data support.

J Proof of the generalized realized-state certificate

Restatement. Fix any realized-state comparator sequence $y_t \in \mathcal{X}_t^+$. Approximate surrogate optimality and the definition of δ_t imply

$$L_t(y_t) - L_t(x_t) \le 2\delta_t + \xi_t = \zeta_t.$$

Since $r_t(z) = L_t(z) + \lambda_t^\top g_t(z, q_t)$,

$$r_t(y_t) - r_t(x_t) \le \zeta_t + \lambda_t^\top g_t(y_t, q_t) - \lambda_t^\top g_t(x_t, q_t).$$

By Lemma 1,

$$-\lambda_t^\top g_t(x_t, q_t) \le \frac{\eta}{2}\|g_t(x_t, q_t)\|_2^2 - \frac{\|\lambda_{t+1}\|_2^2 - \|\lambda_t\|_2^2}{2\eta}.$$

Substitution and summation telescope the queue energy, yielding equation 7 because $\lambda_1 = 0$. Coordinatewise queue recursion gives $\lambda_{T+1} \succeq \eta[\sum_t g_t(x_t, q_t)]_+$ and therefore $\mathrm{Viol}_T \le \|\lambda_{T+1}\|_2/\eta$. □

Corollary 4 (Pointwise-safe data-dependent certificate)**.** *For $y_t = x_t^\star$ in equation 4, $\mathcal{D}_T(y) \le 0$ and*

$$\mathrm{Reg}_T^{\mathrm{tc}} + \frac{\|\lambda_{T+1}\|_2^2}{2\eta} \le \frac{\eta}{2}\sum_{t=1}^T \|g_t(x_t, q_t)\|_2^2 + \sum_{t=1}^T \zeta_t. \tag{11}$$

Pointwise-safe specialization. If $y_t = x_t^\star$, then $g_t(y_t, q_t) \preceq 0$ and $\lambda_t \succeq 0$, hence every summand in $\mathcal{D}_T(y)$ is nonpositive. Dropping it yields Corollary 4.

Comparator strength enters through the measurable $\mathcal{D}_T(y)$. The window-safe theorem controls this term by an observable property of the comparator sequence and learner queue rather than dropping it by pointwise feasibility. This completes the generalized L1 certificate. An L2 policy value additionally requires the state-alignment interface in Proposition 3.

K Window-safe comparator proof

Proof of Theorem 4. Fix one block $B_j = \{a, \dots, b\}$ of length L_j and write $h_t = g_t(y_t, q_t)$. Define prefix pressures $C_t = \sum_{s=a}^t h_s$. Window safety gives $C_b \preceq 0$. Discrete summation by parts yields

$$\sum_{t=a}^b \lambda_t^\top h_t = \lambda_b^\top C_b + \sum_{t=a}^{b-1} (\lambda_t - \lambda_{t+1})^\top C_t.$$

The first term is nonpositive because $\lambda_b \succeq 0$ and $C_b \preceq 0$. Projection in equation 5 is 1-Lipschitz, so $\|\lambda_{t+1} - \lambda_t\|_2 \le \eta\|g_t(x_t, q_t)\|_2 \le \eta B_g$. Also $\|C_t\|_2 \le (t-a+1)B_g$. Hence

$$\sum_{t=a}^{b} \lambda_t^\top h_t \le \eta B_g^2 \sum_{k=1}^{L_j-1} k = \frac{\eta B_g^2}{2} L_j(L_j-1).$$

Summing over blocks proves the first inequality in equation 8; since $L_j \le W$ and $\sum_j L_j = T$, $\sum_j L_j(L_j-1) \le (W-1)T$. Substitution into equation 7 and $\sum_t \|g_t(x_t, q_t)\|_2^2 \le B_g^2 T$ gives equation 9. □

Proposition 5 (The W dependence of comparator debt is order-tight)**.** *For even W dividing T, bounded scalar pressure sequences can satisfy the queue recursion and W-window comparator safety while*

$$\mathcal{D}_T(y) = \eta B_g^2 WT/4.$$

Thus the linear dependence on W in equation 8 is order-tight up to constants under bounded pressure and window safety.

Proof of Proposition 5. Take scalar learner pressure $g_t(x_t, q_t) = B_g$ on every round, so projection is inactive and $\lambda_t = \eta B_g(t-1)$. Let $W = 2m$ and, in every length-W block, set comparator pressure to $-B_g$ for the first m rounds and $+B_g$ for the last m. Each block sums to zero and is therefore window-safe. The common block offset cancels between the two half-block sums, giving debt $\eta B_g^2 m^2 = \eta B_g^2 W^2/4$ per block. With T/W blocks,

$$\mathcal{D}_T(y) = \frac{T}{W}\frac{\eta B_g^2 W^2}{4} = \frac{\eta B_g^2 WT}{4}.$$

Hence the $O(\eta B_g^2 WT)$ dependence is tight up to a constant factor for this information/constraint class. □

For $W = 1$ every prefix sum is a single nonpositive pressure vector, recovering the pointwise result. Larger windows permit individually pressure-producing actions followed by repayment; the W factor quantifies the price of that additional benchmark freedom.

L MINIMAX LOWER BOUND FOR THE EXACT-L1 WINDOW GAME

Restatement and proof plan. Under Definition 3, for every even $W \ge 2$ with $W \mid T$ there is a universal $c > 0$ such that $\mathfrak{M}_{T,W} \ge c\sqrt{WT}$, while $\mathfrak{M}_{T,1} = 0$. We first prove the even divisible statement by Yao's principle, including the $T = W$ boundary case, and then extend the lower bound to every integer $2 \le W \le T$ used in Corollary 3.

Hard subclass and membership in the routing model. Let $W = 2m$ and $N = T/W$. Every round offers a route action a and fallback 0. Their scalar pressures are $g(a) = +1$ and $g(0) = -1$, so any window with exactly m route uses and m fallbacks has zero aggregate pressure. Fallback reward is zero. This is a strict subclass of the main routing model: take one always-feasible route, set $\nu_t = a_t = 0$ so the inventory and realized feasible set are constant, use scalar $x \in \{0, 1\}$, and set $b_t = 1$, $M_t = 2$ in equation 2, giving $g_t(1) = +1$ and $g_t(0) = -1$; route reward is $r_t(x) = \theta_t x$. Hence $\|g_t\|_2 \le 1$, rewards lie in $[0, 1]$, and the fallback satisfies the $-1/2$ margin required by Definition 3. Because the state and candidate set are fixed, the realized-state comparator in this subclass depends only on the obliviously chosen reward sequence, not on the learner's realized trajectory.

Window i has two phases of m rounds. The route reward is p_i in the first phase and p_{i+1} in the second. Let $S_0 = 0$ and sample independent Rademacher signs $\sigma_i \in \{-1, +1\}$ before play. Set $H = 2\sqrt{N}$, $\epsilon = (16\sqrt{N})^{-1}$, and let $A_i = \mathbf{1}\{|S_{i-1}| < H\}$ be the predictable indicator that the walk has not stopped. Set $p_1 = 1/2$ and, for $i = 1, \dots, N$, update $S_i = S_{i-1} + A_i\sigma_i$ and define $p_{i+1} = 1/2 + \epsilon S_i$, so $p_{i+1} - p_i = \epsilon A_i \sigma_i$. Since a stopped walk can overshoot H by at most one, $p_i \in [5/16, 11/16]$ for $i = 1, \dots, N+1$ and all $N \ge 1$. The whole sign sequence is sampled before play, so this is a distribution over deterministic oblivious instances; before every action the learner observes both current rewards and pressures exactly.

A reward-maximizing W-window comparator uses the route during the better half of an active window and fallback during the other half; after stopping, either half is equivalent. All route rewards are positive, while window safety permits at most m route uses because route and fallback pressures are $+1$ and -1. Thus this comparator is the reward-maximizing W-window-safe comparator and has exactly zero pressure in every window.

Regret identity for $N \geq 2$. Normalize the learner's route counts in the two halves of window i by $h_i, b_i \in [0,1]$; h_i is chosen before σ_i is revealed, while b_i may depend on it. Put $u_i = h_i + b_i - 1$ and $U_i = \sum_{j \leq i} u_j$. The learner's total scalar pressure is $2mU_N$, so $\mathrm{Viol}_T = 2m[U_N]_+$. Direct subtraction from the window comparator gives

$$\frac{\mathrm{Reg}_T(y^{(W)})}{m} = -\sum_{i=1}^{N} p_i u_i + \epsilon \sum_{i=1}^{N} A_i e_i, \qquad e_i = \mathbf{1}\{\sigma_i = +1\}(1 - b_i) + \mathbf{1}\{\sigma_i = -1\} b_i.$$

Summation by parts yields

$$-\sum_{i=1}^{N} p_i u_i = -p_N U_N + \epsilon \sum_{i=1}^{N-1} A_i U_i \sigma_i.$$

Therefore

$$\frac{\mathrm{Reg}_T(y^{(W)}) + \frac{1}{2}\mathrm{Viol}_T}{m} = -p_N U_N + [U_N]_+ + \epsilon \sum_{i=1}^{N-1} A_i U_i \sigma_i + \epsilon \sum_{i=1}^{N} A_i e_i. \tag{12}$$

The boundary term is nonnegative because $0 < p_N < 1$: for $U_N \leq 0$ it equals $-p_N U_N$, while for $U_N > 0$ it equals $(1 - p_N)U_N$.

For each active $i < N$, condition on the history before σ_i. Write $z_i = U_{i-1} + h_i - 1$, which is measurable before the sign. If $\sigma_i = +1$, then $U_i = z_i + b_i$ and $U_i\sigma_i + e_i = z_i + b_i + 1 - b_i = z_i + 1$. If $\sigma_i = -1$, then $U_i\sigma_i + e_i = -z_i - b_i + b_i = -z_i$. Hence this cancellation remains valid even though the second-half choice may react to the revealed sign:

$$\mathbb{E}[U_i\sigma_i + e_i \mid \mathcal{F}_{i-1}, A_i = 1] = \tfrac{1}{2}.$$

Dropping the nonnegative boundary and final error terms in equation 12 gives

$$\mathbb{E}\Big[\mathrm{Reg}_T(y^{(W)}) + \tfrac{1}{2}\mathrm{Viol}_T\Big] \geq \frac{m\epsilon}{2}\,\mathbb{E}\left[\sum_{i=1}^{N-1} A_i\right].$$

For the unstopped simple random walk, Doob's maximal inequality applied to S_i^2 gives $\Pr(\max_{i \leq N} |\sum_{j \leq i} \sigma_j| \geq 2\sqrt{N}) \leq 1/4$. Thus with probability at least $3/4$ all first $N-1$ increments remain active, and for $N \geq 2$,

$$\mathbb{E}\Big[\mathrm{Reg}_T(y^{(W)}) + \tfrac{1}{2}\mathrm{Viol}_T\Big] \geq \frac{3m\epsilon(N-1)}{8} \geq \frac{3}{512}\sqrt{WT}.$$

The one-window boundary $N = 1$. Take first-half route reward $1/2$ and second-half reward $1/2 + \delta\sigma$ with $\delta = 1/8$ and $\sigma \in \{-1, +1\}$ sampled before play. Let h be the learner's first-half route fraction and let b_+, b_- be its sign-dependent second-half route fractions. With $u_\pm = h + b_\pm - 1$, the two losses satisfy

$$\frac{\mathrm{Reg}_T + \frac{1}{2}\mathrm{Viol}_T}{m} = \tfrac{1}{2}|u_\pm| + \delta e_\pm, \qquad e_+ = 1 - b_+, \quad e_- = b_-.$$

Averaging the two signs gives

$$\frac{1}{m}\mathbb{E}\big[\mathrm{Reg}_T + \tfrac{1}{2}\mathrm{Viol}_T\big] = \frac{1}{4}(|u_+| + |u_-|) + \frac{\delta}{2}(1 - b_+ + b_-) \geq \frac{\delta}{2},$$

because $|u_+| + |u_-| \geq |b_+ - b_-|$; if $b_+ \geq b_-$ the first term dominates the possible subtraction since $1/4 \geq \delta/2$, and if $b_+ < b_-$ the second term already exceeds $\delta/2$. Hence the expected loss is at least $m/16 = W/32$, a constant fraction of $\sqrt{WT} = W$.

Since $[\mathrm{Reg}_T]_+ \geq \mathrm{Reg}_T$, both cases lower-bound the exact objective $[\mathrm{Reg}_T]_+ + \frac{1}{2}\mathrm{Viol}_T$. The distribution above is fixed before play and consists of deterministic oblivious instances. Therefore Yao's principle applies directly: every randomized learner has a deterministic oblivious instance in the support with expected loss at least the displayed bound.

Arbitrary W and horizons not divisible by W. Fix any integer $2 \le W \le T$, let $W_e = 2\lfloor W/2 \rfloor$ and $N = \lfloor T/W \rfloor \ge 1$, and run the even hard instance above on the first W_e rounds of each of the first N full W-blocks. If W is odd, use the remaining round in each full block as a neutral dummy round; use the same dummy construction on all $T - NW$ trailing rounds. On a dummy round the route has reward 1 and pressure 0, while fallback has reward 0 and pressure -1. This is again a routing subclass, now with $b_t = 1, M_t = 1$ on dummy rounds.

A reward-maximizing target W-window comparator takes the neutral route on every dummy round and uses the even hard comparator on each active subwindow. For odd W, write $W_e = 2m$. If the sole dummy in a full block is taken as the route, window safety permits at most m active route uses; if it is replaced by fallback, the constraint becomes $2r - W_e - 1 \le 0$ and still forces the integer route count $r \le m$. The dummy fallback therefore buys no additional active route and only loses reward 1. The trailing partial block contains only dummy rounds, where the neutral route is pointwise optimal and safe. Hence this comparator is W-window-safe and reward-maximizing for the padded instance.

Let R_a and P_a denote the learner's raw regret and cumulative pressure on active hard rounds, and let k be the number of dummy fallbacks it takes. Padding changes total raw regret and pressure to $R_a + k$ and $P_a - k$. Since $[z - k]_+ \ge [z]_+ - k$ for $k \ge 0$,

$$R_a + k + \tfrac{1}{2}[P_a - k]_+ \ge R_a + \tfrac{1}{2}[P_a]_+ + \tfrac{k}{2} \ge R_a + \tfrac{1}{2}[P_a]_+.$$

Thus dummy rounds cannot reduce the lower-bound quantity, even though the learner may use their negative pressure to offset final violation. Applying the even-instance bound to active horizon $T_a = NW_e$ gives $\Omega(W_e\sqrt{N})$. Since $W_e \ge W/2$ and $T < (N+1)W \le 2NW$,

$$W_e\sqrt{N} \ge \frac{1}{2\sqrt{2}}\sqrt{WT}.$$

This proves the lower half of Corollary 3 for every $2 \le W \le T$, including odd W and $W \nmid T$, with only a universal constant loss. □

The $W = 1$ endpoint. With exact current L1 coefficients, the learner chooses a highest-reward action among those with $g_t(x, q_t) \preceq 0$ at each realized state. This action is exactly a reward-maximizing pointwise-safe comparator on that state and creates no positive pressure, so both regret and violation are zero.

Completion of Corollary 3. For the upper bound in the normalized exact-L1 game, take $B_r = B_g = 1$ and $\sigma = 1/2$ in Corollary 2, set $\gamma = 1/2$, and choose $\eta = \Theta((WT)^{-1/2})$. The resulting pathwise envelope is $C\sqrt{WT}$ for a universal C, and because its regret bound is nonnegative it also bounds $[\mathrm{Reg}_T(y^{(W)})]_+ + \frac{1}{2}\mathrm{Viol}_T$. Taking the supremum over admissible environments, expectation over learner randomization, and then the infimum over learners preserves the same upper bound. The even-divisible construction above, the neutral-padding extension, and the $W = 1$ endpoint give the corresponding lower bound and zero endpoint. This proves the full statement of Corollary 3 with exactly the assumptions of Definition 3. □

Relation to window-budget prior work. The construction is related to the save-or-spend mechanism behind window-budget lower bounds, but the game objects differ. Liakopoulos et al. (2019) use window-feasible K-benchmarks for long-term constraints, while Braverman et al. (2025) obtain $\widetilde{O}(T/\sqrt{W} + \sqrt{WT})$ and a matching lower bound for a pacing benchmark under bandit feedback. In our exact-L1 game all current action rewards and pressures are revealed before acting, so the within-window bandit-learning mechanism behind the $T/\sqrt{W}$ term is absent; the hard construction isolates the $\sqrt{WT}$ save-or-spend term from future allocation uncertainty. Together with the upper bound, this returns to the main claim: after L1 makes the comparator measurable, W governs the unavoidable difficulty of strengthening it.

M WEAK-SAFE VIOLATION BOUND

Corollary 5 (Weak-safe fallback)**.** *If $|r_t(x)| \le B_r$, fallback is weakly safe, and $\|g_t\|_2 \le B_g$, then*

$$\mathrm{Reg}_T^{\mathrm{tc}} \le \frac{\eta B_g^2 T}{2} + \sum_t \zeta_t, \qquad \mathrm{Viol}_T \le \sqrt{B_g^2 T + \frac{2}{\eta}\sum_t \zeta_t + \frac{4B_r T}{\eta}}.$$

With exact coefficients/oracle and $\eta = T^{-1/2}$, this gives $O(\sqrt{T})$ pointwise-safe regret and $O(T^{3/4})$ cumulative violation.

Dropping the nonnegative queue term in equation 11 and using $\sum_t \|g_t\|^2 \le B_g^2 T$ proves the regret part of Corollary 5. Bounded rewards imply each pointwise comparator gap is at least $-2B_r$, so $\mathrm{Reg}_T^{\mathrm{tc}} \ge -2B_r T$. Thus

$$\frac{\|\lambda_{T+1}\|_2^2}{2\eta} \le \frac{\eta B_g^2 T}{2} + \sum_t \zeta_t + 2B_r T.$$

Multiply by 2η, divide the norm bound by η, and use queue dominance to obtain the stated violation inequality. With $\zeta_t = 0$ and $\eta = T^{-1/2}$, the $4B_r T/\eta$ term yields $O(T^{3/4})$. □

N Strict-margin violation bound

Assume exact optimization, $0 \le r_t(x) \le B_r$, and $g_t(0, q_t) \preceq -\sigma \mathbf{1}$. Since $r_t(0) \ge 0$,

$$L_t(x_t) \ge L_t(0) = r_t(0) - \lambda_t^\top g_t(0, q_t) \ge \sigma \|\lambda_t\|_1 \ge \sigma \|\lambda_t\|_2.$$

Because $r_t(x_t) \le B_r$, $\lambda_t^\top g_t(x_t, q_t) \le B_r - \sigma\|\lambda_t\|_2$. Inserting this into Lemma 1 shows that whenever $\|\lambda_t\|_2 \ge B_r/\sigma + \eta B_g^2/(2\sigma)$, squared queue norm cannot increase. One update changes the norm by at most ηB_g, giving the bound in Corollary 1. □

O Window-aware rate frontier proof

Starting from Theorem 4 with exact coefficients/oracle and dropping the nonnegative terminal-queue term gives

$$\mathrm{Reg}_T(y) \le \frac{\eta B_g^2 W T}{2}.$$

The strict-margin queue bound in Corollary 1 and queue dominance imply

$$\mathrm{Viol}_T \le \frac{\|\lambda_{T+1}\|_2}{\eta} \le \frac{B_r}{\sigma\eta} + \frac{B_g^2}{2\sigma} + B_g.$$

This proves equation 10. For a weight $\gamma > 0$, the only η-dependent part of the resulting upper envelope is

$$f(\eta) = \frac{B_g^2 W T}{2}\eta + \frac{\gamma B_r}{\sigma\eta}.$$

It is strictly convex on $\eta > 0$ and is minimized when the two derivative terms balance,

$$\eta_\gamma^* = \sqrt{\frac{2\gamma B_r}{\sigma B_g^2 W T}}.$$

Substitution gives the weighted bound stated in Corollary 2. Equivalently, the arithmetic–geometric mean inequality gives the same minimum directly. Thus the window factor can be allocated between comparator regret and learner violation through the single queue stepsize rather than being attached only to the regret term. □

The result optimizes the proved CVQ certificate envelope with respect to η. Proposition 5 separately shows that the W factor in queue-weighted comparator debt is order-tight up to constants under bounded pressure and window safety. This gives the controller-facing consequence of the frontier without turning η into an information axis.

P Algorithm and route computation

Algorithm 1 gives the executable input–score–decision–update–audit loop in the main text. Here we record the route filtering details used by the released implementation. The implementation has a fixed static edge universe and pre-enumerates a route catalog for each origin–destination (OD) pair. At each round, routes containing non-operational edges are removed, fallback is appended, and remaining actions are scored. This enforces selection only from routes present in the realized graph. The algorithmic steps, information access, and update equations are otherwise unchanged.

Q COMPUTATIONAL REGIMES AND ORACLE ERROR

With an explicit catalog of K_t paths, exact maximization is finite enumeration. For a directed acyclic graph (DAG), additive reduced lengths can be optimized over all directed paths by topological dynamic programming even if some lengths are negative. For cyclic graphs, standard shortest-path correctness requires no relevant reachable negative-cost cycle; unrestricted negative-cycle shortest *simple* path lies outside this exact-solver regime. Whenever the controller sees only a truncated catalog or an approximate solver, the reference audit set remains the larger realized catalog and

$$\xi_t = \sup_{x \in \mathcal{X}_t^+} \widehat{L}_t(x) - \widehat{L}_t(x_t)$$

measures the resulting optimization loss. This is precisely the quantity used in the limited-route-search sensitivity and certificate-validity experiments.

R EXPERIMENTAL EVIDENCE MATRIX AND COMPLETE PROTOCOL

All figures and tables are generated from the result CSVs without smoothing or post-hoc changes to the estimates. Normal intervals are $1.96s/\sqrt{n}$; they are descriptive rather than simultaneous or dependence-robust intervals. Public and robustness plots display means, with companion SDs retained in the tables and CSVs. Chronological report weeks are treated as ordered report units rather than modeled as independent draws. Displayed zero violations remain zero (symlog is only an axis transform), and the seed-0 qualitative trace is a fixed diagnostic rather than a selected average.

This appendix records the full protocol used for every empirical statement in the main paper. All reported result CSVs are generated by the released implementation and matched to the tables and figures below.

Table 6: Claim–evidence map for the rerun.

Claim	Primary evidence	Boundary / control
Window comparators become stronger with repayment horizon	Exact blockwise mixed-integer optimization, $W = 1, 2, 4, 8, 16, 32$	Eight-route L1 catalog, $T = 400$; off-trajectory value is handled by the separate reset replay
Pathwise certificate charges observable approximation error	Forecast-noise and limited-search sweeps	Same audited candidate set for the search-error sweep; all coefficients stored post hoc
Reserve price and fallback control depletion debt	NoReserve and NoReject ablations	Scarcity regime chosen before report seeds
Behavior under nontrivial shifts	Burst, trust-drop, replenishment, topology sweeps	Axis-specific stress suite with rankings reported separately for each stress
Mechanism under public routing inputs	Abilene weeks 13–24 + measured QKD SKR ratios	Empirical traffic/SKR inputs inside a controlled reset replay

R.1 SYNTHETIC ENVIRONMENT AND PAIRED RANDOMNESS

The base environment is a six-layer directed acyclic graph with width 4, edge probability .68, and up to 32 shortest-simple-path candidates per source–destination pair. The inventory maximum is 96, initial inventory is 72 (multiplied by .85 on the top 20% edge-betweenness "critical" edges), and the reserve floor is 30. Base consumption is 1, doubled on burst rounds; base/critical mean replenishment is .25/.13. Trust-edge removal occurs with probability .02 per round, and the base burst probability is .12. The two pressure coordinates are critical-route use relative to budget .35 and bounded additive reserve exposure relative to budget .06; fallback therefore has strictly negative pressure $(-.35, -.06)$. Reward is clipped to $[0, 1]$ and trades path quality, demand service, path length, and critical-edge use. All exogenous requests, trust masks, replenishment, edge quality, and forecast perturbations are generated before actions for a given seed, so paired methods receive identical exogenous sequences.

Report seeds are 0–11. Hyperparameters are selected only from validation seeds 100–103 at $T = 800$. Main horizons are $200, 400, 800, 1600$. Confidence intervals are mean $\pm 1.96s/\sqrt{n}$ across paired

report seeds/weeks; deterministic pathwise certificate checks are reported as pass counts, not statistical tests.

R.2 Equal-budget baseline tuning

Each parameterized family receives five settings. The selection rule maximizes mean validation reward among settings with mean $\mathrm{Viol}_T/\sqrt{T} \le .20$; if none passes, it chooses minimum violation. The frozen selections are:

Table 7: Validation grid and frozen reported setting.

Method	Grid	Selected
CVQ-Forecast	$c \in \{4, 8, 12, 16, 24\}$ in $\eta = c/\sqrt{T}$	16
Decaying-PD	$\{.4, .8, 1.2, 1.6, 2.4\}$	2.4
AdaGrad-PD	$\{.2, .4, .8, 1.2, 1.6\}$	.8
EG-PD	$\{.2, .4, .8, 1.2, 1.6\}$	1.2
FixedPenalty	$\{.25, .5, 1, 2, 4\}$	4
SafeGreedy / RewardOnly	no tuned hyperparameter	–

No report seed is used for tuning. Decaying-PD and FixedPenalty do not reach the .20 validation threshold in the declared grid, so the rule selects their minimum-violation endpoints as specified in advance.

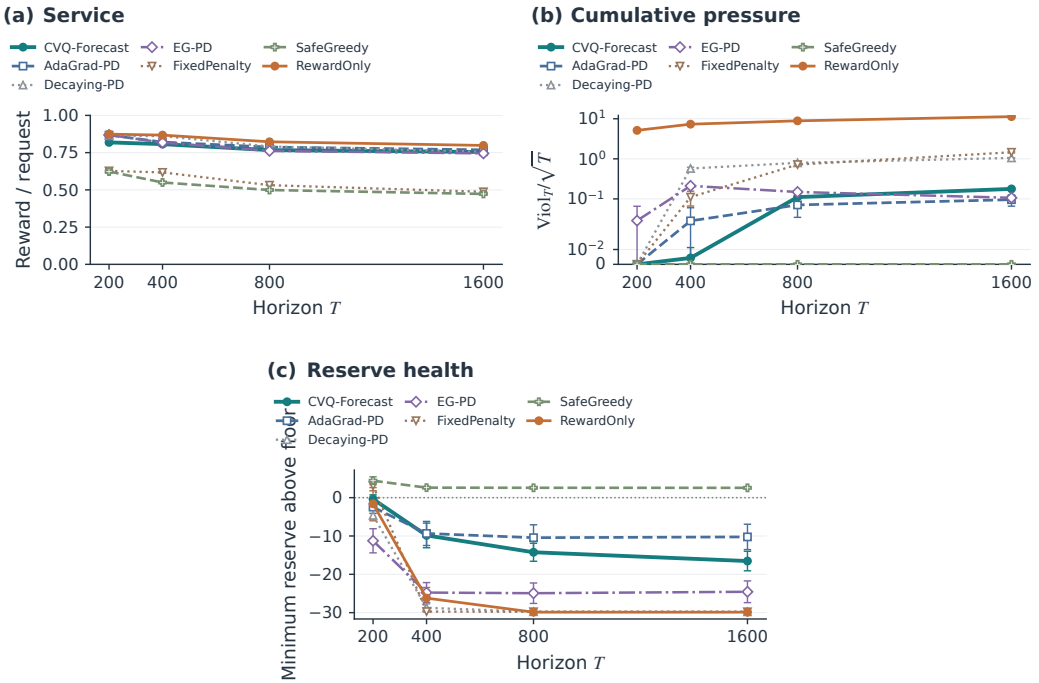

Figure 7: Complete seven-method horizon results redrawn from the original summaries; all four horizons and all methods, original normal 95% intervals. The violation axis is symlog with linear threshold .03 and includes zero.

Table 8: **Synthetic endpoint at $T = 800$.** Mean $\pm$ normal 95% interval over 12 paired report seeds. Negative minimum reserve records a crossing below the diagnostic reserve floor; physical route feasibility is still enforced. Drop rate is the fraction of requests sent to fallback/rejection.

Method	Reward/request	$\mathrm{Viol}_T/\sqrt{T}$	Min. reserve	Drop rate
CVQ-Forecast	0.767 ± 0.003	0.111 ± 0.028	-14.25 ± 2.35	0.135 ± 0.003
AdaGrad-PD	0.787 ± 0.003	0.071 ± 0.036	-10.44 ± 3.37	0.112 ± 0.003
Decaying-PD	0.791 ± 0.006	0.793 ± 0.065	-29.76 ± 0.05	0.107 ± 0.007
EG-PD	0.762 ± 0.004	0.149 ± 0.005	-24.93 ± 2.67	0.141 ± 0.004
FixedPenalty	0.532 ± 0.002	0.719 ± 0.067	-29.77 ± 0.01	0.405 ± 0.002
SafeGreedy	0.499 ± 0.001	0.000 ± 0.000	2.57 ± 0.01	0.443 ± 0.001
RewardOnly	0.823 ± 0.007	8.878 ± 0.240	-29.87 ± 0.01	0.068 ± 0.008

R.3 Full $T = 800$ Endpoint Comparison

Table 8 reports the complete nominal synthetic endpoint after hyperparameter selection on the disjoint validation seeds in the preceding subsection. All methods are evaluated on the same 12 paired report seeds.

The endpoint shows that the baseline methods occupy different reward–violation–reserve operating points. AdaGrad-PD has higher reward and lower normalized violation than the reported fixed-stepsize CVQ operating point at $T = 800$, whereas RewardOnly and SafeGreedy expose the high-service and pointwise-safe extremes. The main-text experiments therefore focus on the certificate terms, comparator-strength mechanism, and L1–L2 state-alignment distinction rather than on this single nominal endpoint.

R.4 Full $T = 800$ endpoint table

Table 9: Synthetic endpoint at $T = 800$, mean $\pm$ 95% CI over 12 report seeds.

Method	Reward	$\mathrm{Viol}_T/\sqrt{T}$	Min reserve	Drop rate
CVQ-Forecast	$.767 \pm .003$	$.111 \pm .028$	-14.25 ± 2.35	$.135 \pm .003$
AdaGrad-PD	$.787 \pm .003$	$.071 \pm .036$	-10.44 ± 3.37	$.112 \pm .003$
Decaying-PD	$.791 \pm .006$	$.793 \pm .065$	$-29.76 \pm .05$	$.107 \pm .007$
EG-PD	$.762 \pm .004$	$.149 \pm .005$	-24.93 ± 2.67	$.141 \pm .004$
FixedPenalty	$.532 \pm .002$	$.719 \pm .067$	$-29.77 \pm .01$	$.405 \pm .002$
SafeGreedy	$.499 \pm .001$	0	$2.57 \pm .01$	$.443 \pm .001$
RewardOnly	$.823 \pm .007$	$8.878 \pm .240$	$-29.87 \pm .01$	$.068 \pm .008$

The endpoint clarifies the role of the empirical comparison. AdaGrad-PD attains higher reward, lower normalized violation, higher minimum reserve, and lower drop rate than the fixed-stepsize CVQ operating point at $T = 800$. RewardOnly and SafeGreedy expose the opposite extremes of immediate service and pointwise safety. We therefore use the experiments to study the certified tradeoff, approximation charges, and comparator behavior rather than to select a universally dominant optimizer.

R.5 Window-safe comparator strength

For this diagnostic, we use $T = 400$ and a declared eight-route finite catalog to make exact blockwise mixed-integer optimization tractable on all 12 seeds. For each block, binary variables select exactly one audited candidate per round and enforce both aggregate pressure coordinates. The objective is total audited reward; CVQ scores and queue prices do not enter the comparator optimization.

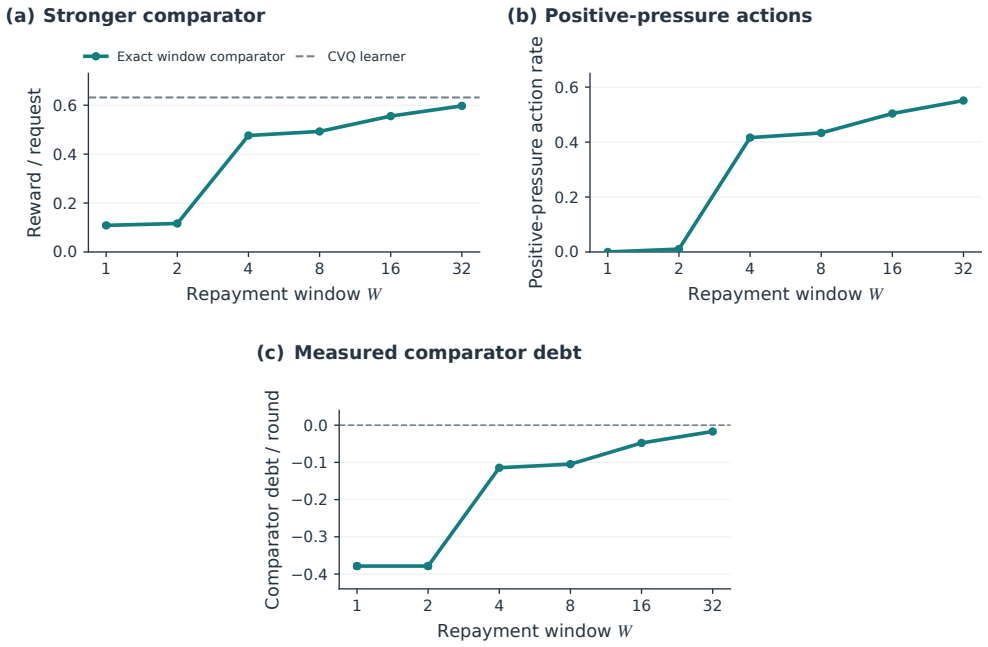

Figure 8: Exact W-window comparator reward, individually positive-pressure action rate, and measured queue-weighted comparator debt. Dashed lines show CVQ reward and zero debt.

Table 10: Exact window-safe comparators, mean $\pm$ 95% CI over 12 report seeds. The window certificate passes for every seed at every W.

W	Comparator reward	Positive-pressure rate	Debt/round	Pass
1	$.109 \pm .008$	0	$-.379 \pm .007$	12/12
2	$.116 \pm .007$	$.010 \pm .002$	$-.379 \pm .007$	12/12
4	$.476 \pm .006$	$.416 \pm .007$	$-.114 \pm .004$	12/12
8	$.493 \pm .005$	$.433 \pm .006$	$-.105 \pm .005$	12/12
16	$.555 \pm .004$	$.504 \pm .006$	$-.048 \pm .004$	12/12
32	$.597 \pm .005$	$.551 \pm .007$	$-.017 \pm .005$	12/12

CVQ reward on these same realized trajectories is $.632 \pm .005$. The diagnostic targets the comparator hierarchy: larger repayment windows admit more individually pressure-producing actions, raise the hindsight comparator value, and move audited debt toward zero while the generalized certificate remains valid.

R.6 Targeted scarcity ablation

The ablation lowers replenishment before running any report seed.

Table 11: Scarcity ablation, $T = 800$, mean $\pm$ 95% CI.

Method	Reward	$\mathrm{Viol}_T/\sqrt{T}$	Min reserve	Drop
CVQ-Forecast	$.672 \pm .002$	$.169 \pm .012$	$-29.84 \pm .01$	$.242 \pm .002$
CVQ-Oracle	$.669 \pm .002$	$.186 \pm .009$	$-29.84 \pm .01$	$.249 \pm .002$
CVQ-NoReserve	$.677 \pm .003$	$1.352 \pm .030$	$-29.89 \pm .01$	$.235 \pm .003$
CVQ-NoReject	$.786 \pm .007$	$3.051 \pm .211$	$-29.91 \pm .01$	$.109 \pm .008$

The close Oracle and Forecast operating points suggest that forecast error is not the dominant source of degradation relative to the reserve/fallback mechanisms in this scarcity regime. Removing the reserve coordinate or fallback increases accumulated pressure far more than it changes reserve at the

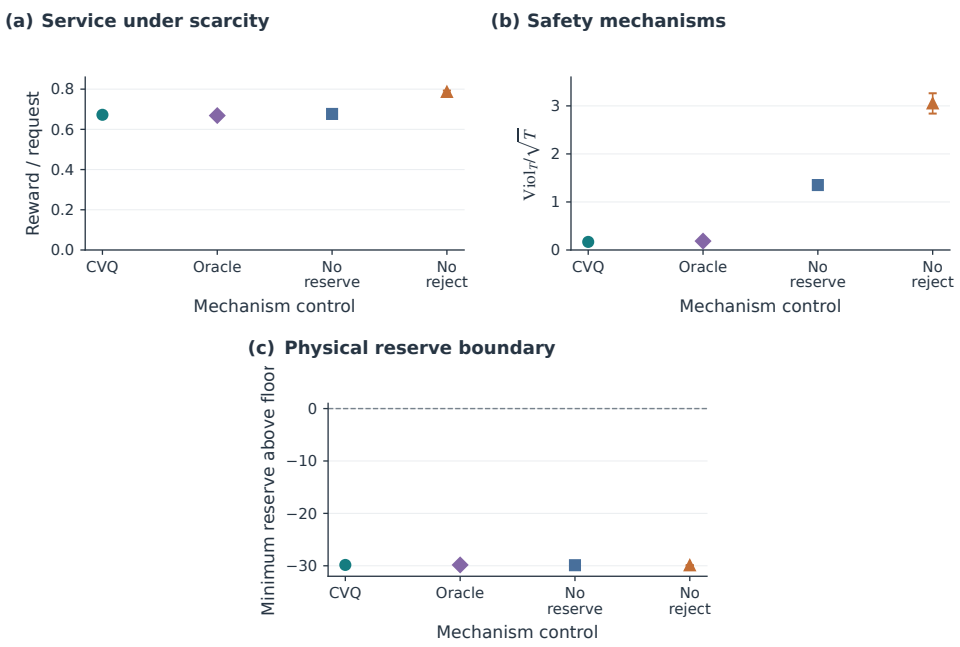

Figure 9: Scarcity ablation: reward, normalized violation, and minimum reserve (12 seeds).

eventual physical floor, which is why we interpret the ablation through the long-run constraint rather than through final inventory alone.

R.7 STEPSIZE AND ERROR-CHANNEL SENSITIVITY

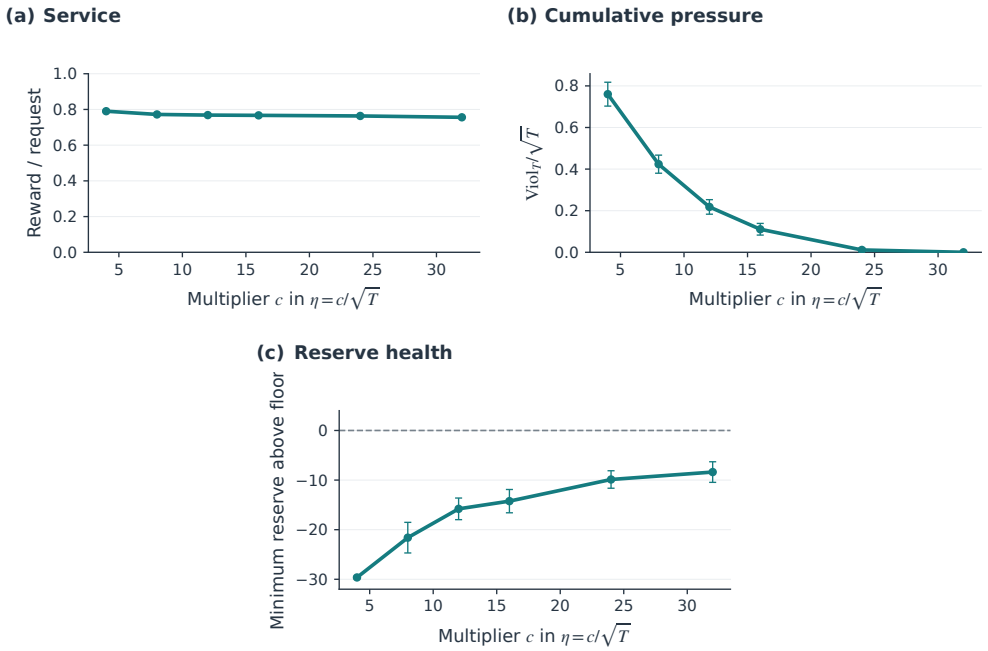

Figure 10: Fixed-stepsize CVQ frontier at $T = 800$, with $\eta = c/\sqrt{T}$ and no retuning on report seeds.

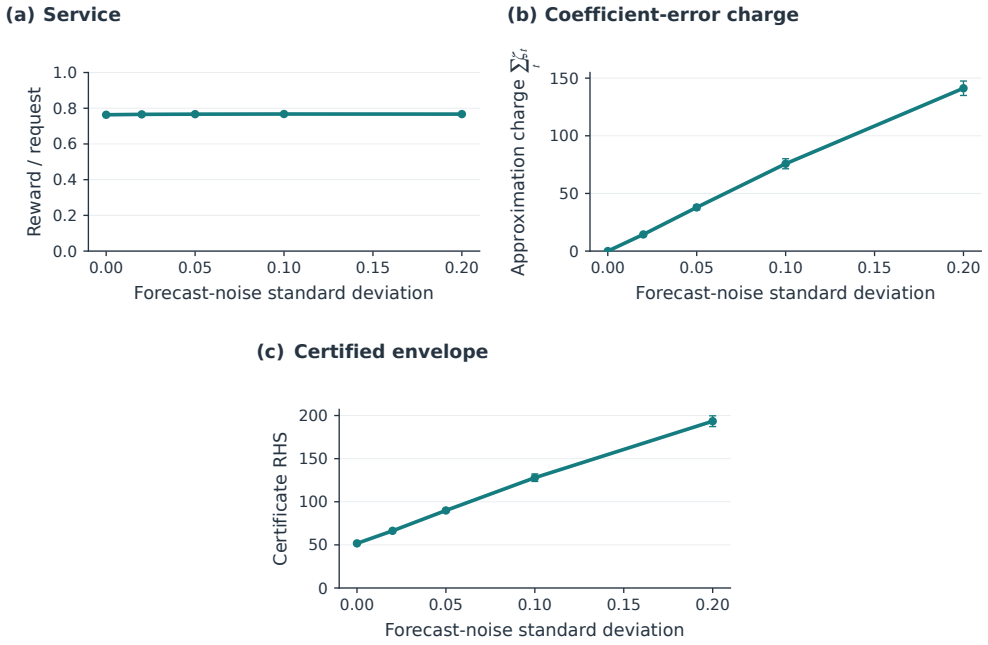

Figure 11: Coefficient misspecification: forecast noise enlarges the measured $\sum_t \zeta_t$ term and hence the pathwise certificate RHS.

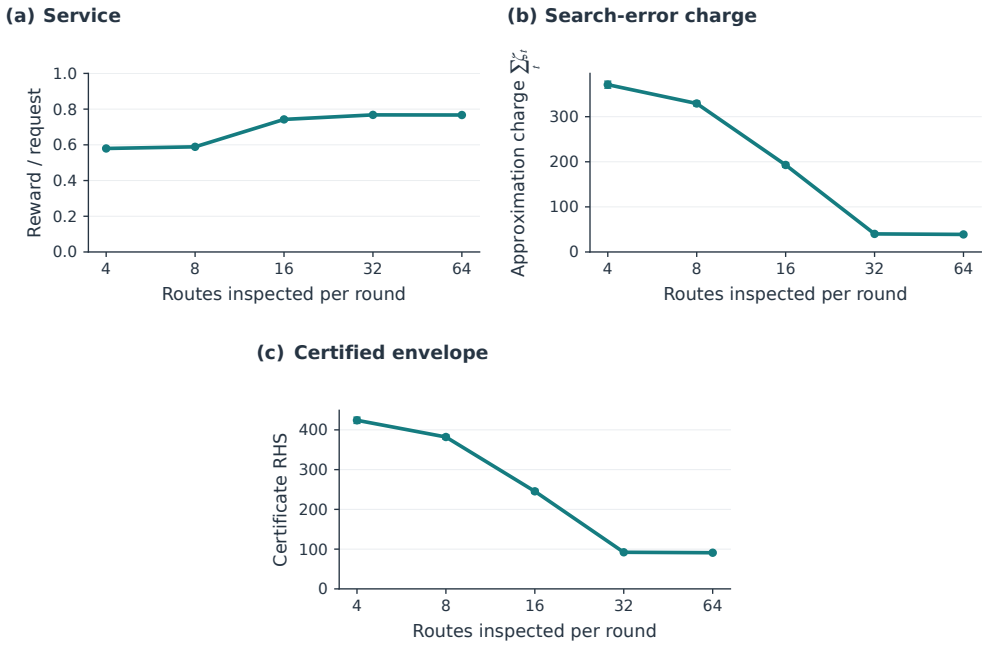

Figure 12: Route-search approximation with a fixed 64-route audited L1 set. The controller inspects only the indicated number of nonfallback routes; ξ_t is computed against the full audited set.

Table 12: Selected error-channel points at $T = 800$ (12 seeds). Pass is the fraction satisfying the pointwise pathwise certificate.

Axis	Setting	Reward	$\sum_t \zeta_t$	Cert. RHS	Pass
Forecast s.d.	0	.763	0.0	51.7	12/12
	.05	.767	37.9	89.9	12/12
	.20	.767	141.2	193.4	12/12
Search budget	4	.580	371.0	424.1	12/12
	16	.742	192.9	245.4	12/12
	32	.768	40.1	92.1	12/12
	64	.767	38.9	90.9	12/12

The search experiment differs importantly from changing the declared candidate set: all 64 audited candidate values remain available post hoc, but the controller is intentionally allowed to inspect only the first K feasible routes. Therefore the resulting score gap is the theorem's measurable ξ_t, rather than an unobserved missing-action effect.

R.8 Robustness axes

Robustness is defined only along the four tested axes. Each extended sweep uses eight paired seeds and keeps the frozen main hyperparameters. Rankings vary across the tested stresses. For example, severe bursts lower reward and increase rejection for both CVQ and AdaGrad, while zero-pressure SafeGreedy sacrifices substantially more service. Stronger replenishment can reduce aggregate violation without preventing local reserve excursions, which is why the paper reports both pressure and minimum inventory.

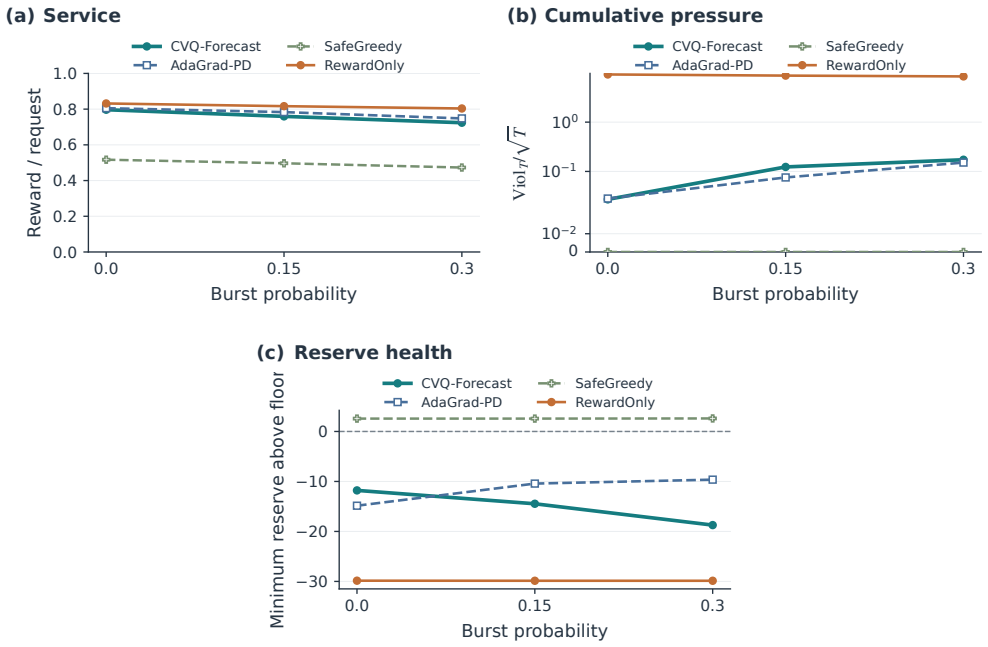

Figure 13: Demand-burst stress: reward, normalized violation, and minimum reserve.

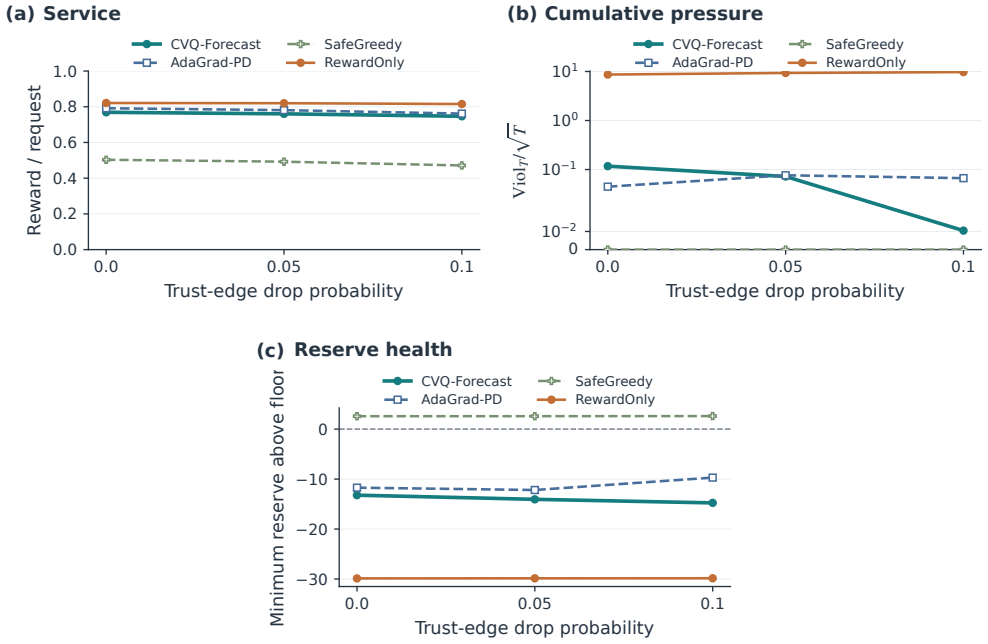

Figure 14: Trust-edge-removal stress.

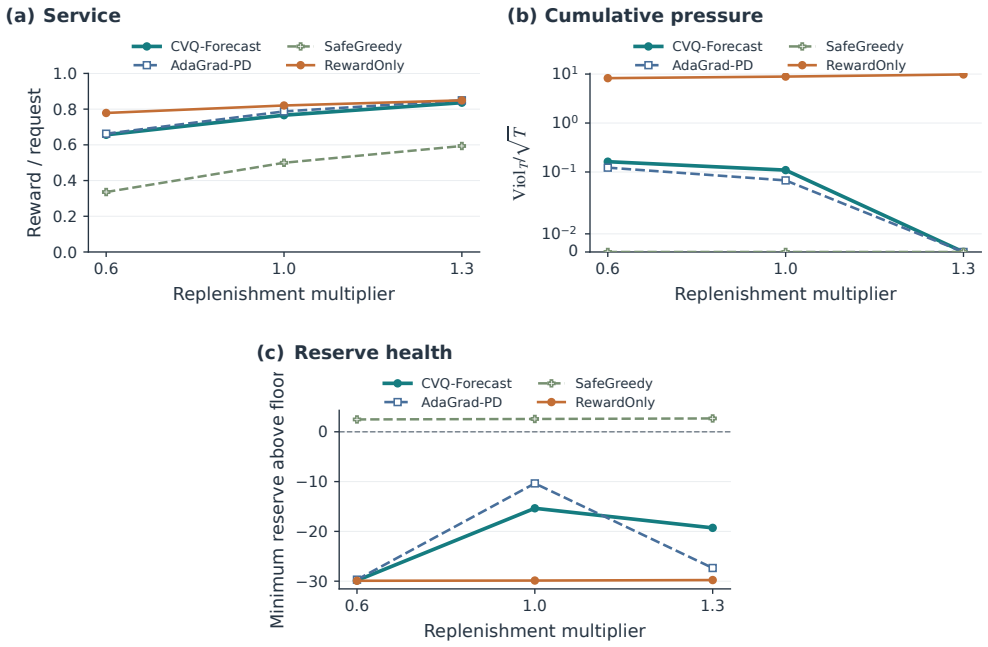

Figure 15: Replenishment scaling stress.

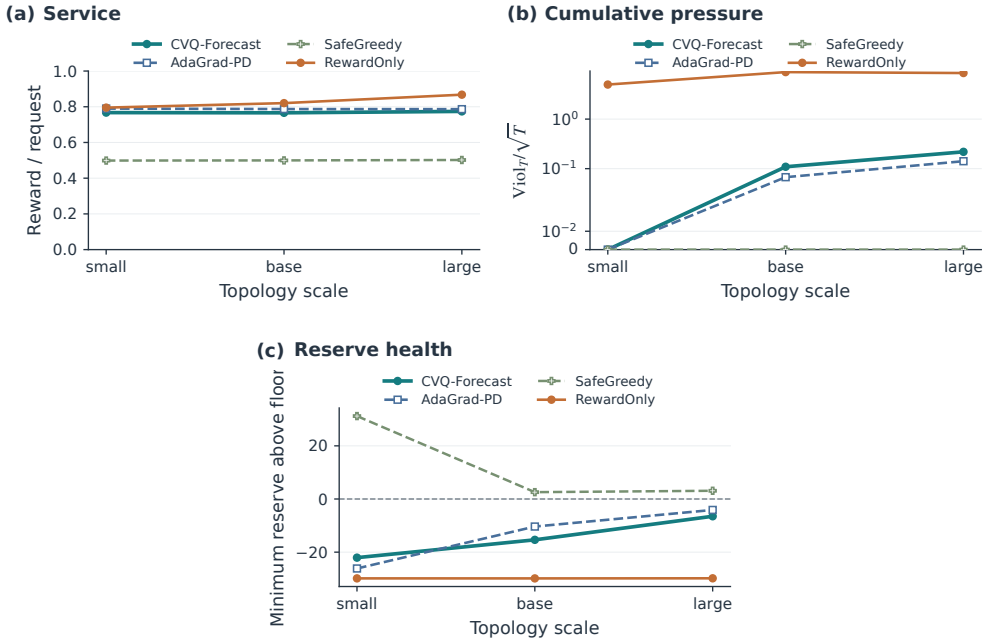

Figure 16: Topology-scale stress.

R.9 Efficiency and implementation cost

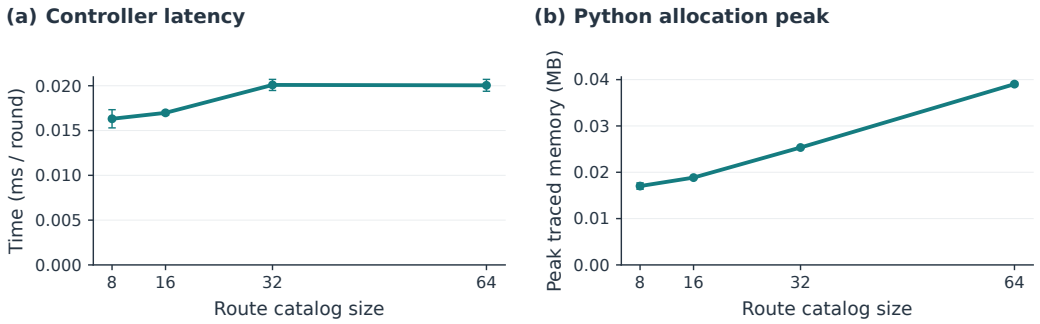

Figure 17: CVQ Python controller time and peak traced Python memory versus route-catalog size ($T = 400$, three seeds).

At a 32-route catalog the six measured controllers use: CVQ .0214 ms/round and .0304 MB; AdaGrad .0203 ms and .0254 MB; EG .0204 ms and .0255 MB; FixedPenalty .0182 ms and .0253 MB; SafeGreedy .0251 ms and .0253 MB; RewardOnly .0060 ms and .0253 MB. CVQ catalog scaling from 8 to 64 routes changes time from .0163 to .0200 ms/round and traced memory from .0170 to .0390 MB. The measurements include Python overhead and are intended for within-code comparisons only.

S Certificate validity and error-channel diagnostics

The certificate is deterministic conditional on the realized audited path, so we verify it seed by seed rather than infer it from average performance. In all reported forecast-noise, route-search-budget, and exact-window experiments, the corresponding inequality holds on every seed. The empirical role of this check is to decompose degradation into measured coefficient mismatch δ_t, route-search suboptimality ξ_t, and comparator debt. Forecast noise from 0 to .20 raises mean $\sum_t \zeta_t$ from 0 to 141.2; restricting a 64-route audited set to four inspected routes raises it to 371.0. In both cases the certificate RHS expands by the explicit amount recorded in the trace. This is the mechanism explicitly charged by Equation 6 and Theorem 3.

T Public-data controlled replay and QKD-native baselines

The public replay was built from the two source archives documented below. The supplement retains the processed requests and outcomes, and a checksum-verified acquisition script supplies the four QKD measurement files needed to rerun the public replay.

Abilene conversion. The source archive contains the 12-node Internet2/Abilene topology with 30 directed internal links and 24 weekly files `X01.gz`–`X24.gz`. Each week contains 2016 five-minute matrices, for 48,384 matrices in total. Every source line stores a 144×5 record; the converter reshapes the first value of each OD record into a 12×12 measured traffic matrix, removes the diagonal, and deterministically samples one OD request per frame with probability proportional to its measured positive traffic. Demand is normalized by the global positive median of the sampled traffic. This scale uses all 24 weeks, including report weeks, so the normalization is transductive rather than training-only. All 24 weeks are parsed; weeks 13–24 form 12 report replicates, and every controller receives the identical request trace within a week. This sampling rule preserves chronology and uses more of the measured traffic distribution than selecting a single largest OD pair per snapshot.

Measured QKD calibration. The QKD V2 archive contains coexistence and characterization sweeps with QBER and SecureKeyRate measurements. We evaluate three physical calibrations:

the 50-km no-injected-signal coexistence baseline, 12-dBm coexistence, and random loss perturbation. For each condition, positive SKR is divided by the median baseline SKR at the *same* VOA attenuation; the resulting empirical ratio distribution is clipped to $[.02, 4]$ and sampled with a fixed per-week calibration seed. Thus laboratory sweep order is never reinterpreted as deployment time. The mean/median multipliers are approximately $.995/1.000$ for coexistence baseline, $.909/.916$ for 12 dBm, and $.743/.744$ for random perturbation. These multipliers scale controlled per-edge replenishment.

QKD-native controls. ShortestAvailable-QKD selects a minimum-hop currently feasible route. WidestShortest-QKD first restricts to minimum-hop routes and chooses the one with largest bottleneck residual inventory, matching directly executable resource-aware logic from the routing literature (López et al., 2026). These controls have no access to CVQ pressure prices.

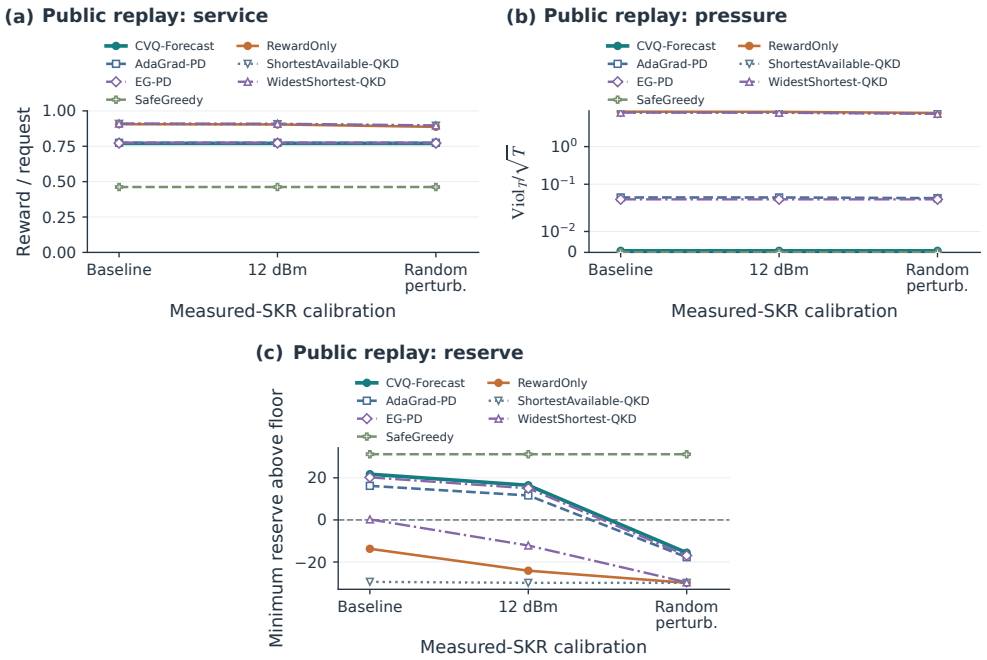

Figure 18: Public Abilene/QKD controlled replay across three measured-SKR calibration regimes; means over 12 chronological report weeks.

Table 13: Public replay, mean ± 95% CI over 12 report weeks. Drop rates are means.

Condition	Method	Reward	$\mathrm{Viol}_T/\sqrt{T}$	Min reserve	Drop
Baseline	CVQ-Forecast	$.770 \pm .008$	$.001 \pm .001$	21.72 ± 7.49	.159
	AdaGrad-PD	$.774 \pm .008$	$.045 \pm .015$	16.18 ± 9.05	.156
	EG-PD	$.772 \pm .008$	$.040 \pm .012$	20.21 ± 9.45	.157
	SafeGreedy	$.462 \pm .009$	0	$31.20 \pm .00$	.507
	RewardOnly	$.906 \pm .007$	$8.463 \pm .339$	-13.71 ± 12.49	.006
	ShortestAvailable-QKD	$.908 \pm .007$	$8.019 \pm .283$	$-29.41 \pm .89$	.008
	WidestShortest-QKD	$.911 \pm .007$	$8.012 \pm .300$	$.25 \pm 14.14$	.006
12 dBm	CVQ-Forecast	$.770 \pm .008$	$.001 \pm .001$	16.45 ± 9.61	.159
	AdaGrad-PD	$.774 \pm .008$	$.045 \pm .015$	11.63 ± 11.69	.156
	EG-PD	$.773 \pm .008$	$.040 \pm .012$	15.02 ± 12.05	.157
	SafeGreedy	$.462 \pm .009$	0	$31.20 \pm .00$	.507
	RewardOnly	$.904 \pm .008$	$8.376 \pm .360$	-24.12 ± 6.53	.008
	ShortestAvailable-QKD	$.906 \pm .008$	$7.926 \pm .302$	$-29.89 \pm .01$	.010
	WidestShortest-QKD	$.909 \pm .008$	$7.951 \pm .317$	-12.15 ± 11.93	.007
Random pert.	CVQ-Forecast	$.770 \pm .008$	$.001 \pm .001$	-15.57 ± 7.91	.160
	AdaGrad-PD	$.774 \pm .008$	$.043 \pm .015$	-17.72 ± 7.29	.156
	EG-PD	$.772 \pm .009$	$.040 \pm .012$	-16.92 ± 7.61	.157
	SafeGreedy	$.462 \pm .009$	0	$31.20 \pm .00$	.507
	RewardOnly	$.887 \pm .012$	$7.819 \pm .393$	$-29.88 \pm .02$	.026
	ShortestAvailable-QKD	$.896 \pm .011$	$7.466 \pm .389$	$-29.90 \pm .00$	.021
	WidestShortest-QKD	$.897 \pm .011$	$7.436 \pm .428$	$-29.64 \pm .45$	.020

The replay separates immediate service from endogenous resource pressure: the QKD-native and reward-only rules provide more immediate service but accumulate much larger pressure. Random measured perturbations can also drive minimum reserve below the controlled floor even when aggregate pressure is near zero, showing that the cumulative-pressure metric and a hard reserve floor are distinct constraints.

L1 versus L2 auditability stress. We use the public replay to instantiate the distinction in Figure 1. CVQ-Forecast generates one logged trajectory. For the stateless target rules RewardOnly, SafeGreedy, ShortestAvailable-QKD, and WidestShortest-QKD, the L1 score $\bar{V}_{\mathrm{L1}}(\pi \mid A)$ evaluates the target action on each CVQ-realized state without changing the log. We then reset the controlled simulator with the same Abilene request sequence, measured-SKR multiplier sequence, trust/quality seed, and 24-route catalog, and execute the target itself to obtain $\bar{V}_{\mathrm{L2}}(\pi)$. The targeted scarcity stress changes only controlled resource parameters: initial/max inventory $30/42$, reserve floor 18, critical initial factor $.80$, base/critical replenishment $.020/.010$, and base consumption 1.50. It is designed to activate endogenous state divergence and is not used for controller selection.

Table 14: Same-state L1 score versus own-trajectory L2 reward on the public coexistence-baseline case, mean ± 95% CI over 12 held-out weeks. "Gap" is L1 minus L2.

Regime	Target rule	L1 reward	L2 reward	Gap
Nominal	RewardOnly	$.911 \pm .001$	$.906 \pm .007$	$.004 \pm .007$
	ShortestAvailable-QKD	$.915 \pm .001$	$.908 \pm .007$	$.007 \pm .008$
	WidestShortest-QKD	$.915 \pm .001$	$.911 \pm .007$	$.004 \pm .008$
	SafeGreedy	$.462 \pm .009$	$.462 \pm .009$	$-.000 \pm .001$
Scarcity	RewardOnly	$.612 \pm .021$	$.278 \pm .007$	$.334 \pm .017$
	ShortestAvailable-QKD	$.614 \pm .021$	$.295 \pm .009$	$.319 \pm .016$
	WidestShortest-QKD	$.614 \pm .021$	$.296 \pm .009$	$.318 \pm .016$
	SafeGreedy	$.059 \pm .004$	$.220 \pm .005$	$-.160 \pm .003$

The nominal regime is a useful sanity check: when resource feedback is weak, the same-state and own-trajectory quantities are close for the aggressive routing rules. Scarcity activates the mechanism. The same-state CVQ log preserves many routes that RewardOnly/Shortest/Widest would deplete on their own, so their L1 scores substantially overstate L2 reward; SafeGreedy goes in the opposite direction because the CVQ trajectory reaches states on which pointwise safety rejects more aggressively. The sign reversal shows that the L1-to-L2 gap is policy- and trajectory-dependent in this replay. The case therefore illustrates the information distinction under empirical network inputs, while the routing objective and reset semantics remain those of the controlled study.



Figure 19: Public Abilene/QKD scarcity case: L1 same-state reward on the CVQ log versus L2 reset replay on each target rule's own trajectory (12 weeks, mean and 95% CI). Controlled reset replay supplies the alternative-state reconstruction required for L2.

Provenance and redistribution. The QKD archive includes a CC BY 4.0 license. The released code records the uploaded archive hashes and per-source-file hashes. The final code package ships converters, processed manifests, and exact commands for regenerating the raw-derived artifacts from the source archives.

U BOUNDARY REGIMES AND QUALITATIVE INTERPRETATION

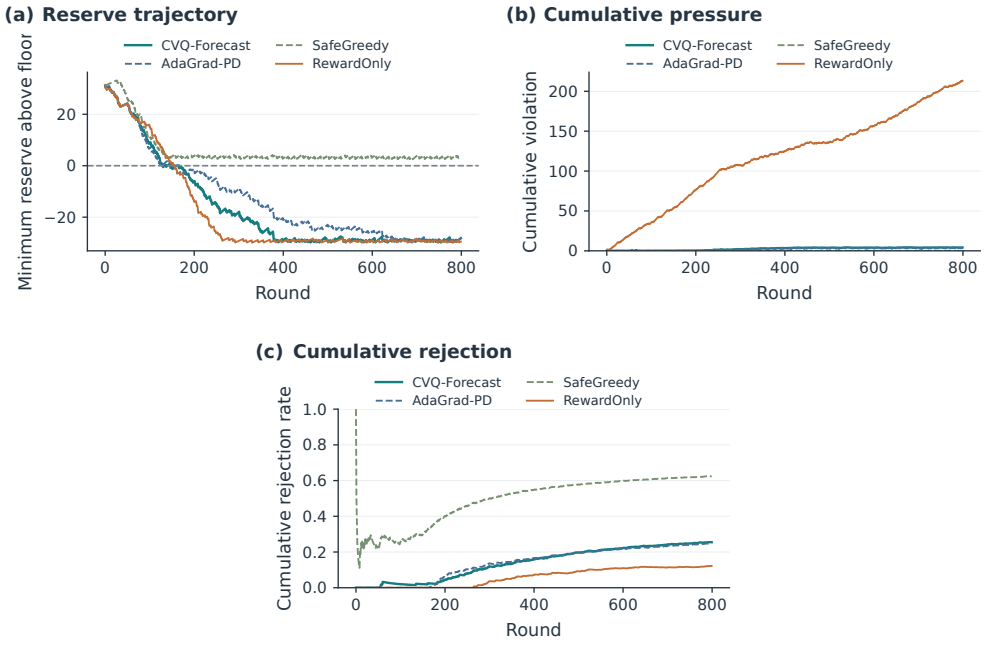

Figure 20: Fixed seed-0 scarcity trajectory: pre-decision minimum reserve, cumulative violation, and cumulative rejection rate.

Physical scarcity. When replenishment is insufficient, no controller can manufacture key/inventory capacity. SafeGreedy rejects heavily to preserve reserve; RewardOnly continues serving and depletes inventory; CVQ/AdaGrad interpolate through pressure prices. A negative minimum reserve means inventory has fallen below the diagnostic reserve floor, not that an infeasible route was executed: physical feasibility is enforced against nonnegative required inventory, while the reserve floor is an early-warning/safety quantity.

Route-search approximation. With a 64-route audited L1 set but only four inspected routes, reward falls to .580 and $\sum_t \zeta_t$ reaches 371.0. Because the uninspected candidate coefficients are still present in the post-hoc audit, this is a genuine measurable ξ_t approximation error. If those candidate outcomes were absent from the record entirely, those actions would instead fall under the L0 support boundary and would require additional logging or outcome modeling.

Coefficient misspecification. Increasing forecast noise expands δ_t and the certificate RHS, while small perturbations can change route selection in either direction. We therefore report the mechanism quantity $\sum_t \zeta_t$ directly; it is the quantity linked to the theorem.

V Reproducibility, compute, code map, and resources

The released runs use the following software/hardware environment and code paths.

Execution environment. Runs used Linux 6.18.44 (x86-64), five exposed virtual CPUs on an AMD EPYC 9V74, Python 3.13.5, NumPy 2.3.5, pandas 2.2.3, NetworkX 3.6.1, SciPy 1.17.0, and Matplotlib 3.10.8. Controller time uses `time.perf_counter` and peak Python memory uses `tracemalloc`; both include Python overhead and support only within-implementation comparisons.

Code map. Core modules are `auditable_routing/envs.py`, `controllers.py`, `evaluation.py`, `comparator.py`, `public_data.py`, and `public_env.py`. Reproduction scripts are `run_tuning_parallel.py`, `run_main.py`, `run_ablation.py`, `run_sensitivity.py`, `run_theory_diagnostics.py`, `run_robustness.py`, `run_public_replay.py`, `run_audit_gap.py`, `run_efficiency.py`, `run_qualitative.py`, and `plot_all.py`. The top-level `run_all.sh` documents the full rerun order.

Raw outputs and statistical protocol. Raw and aggregated CSVs are separate; no best report seed is selected. Forecast-based methods share common-random-number surrogate noise, validation/report seeds are disjoint, and the full validation grid is retained. Certificate outputs store δ_t, ξ_t, ζ_t, LHS/RHS/slack, and pass indicators; exact window-comparator outputs retain every seed and W used in the tables. Extended robustness uses eight paired seeds and is explicitly labeled as an appendix stress suite.

Public assets. The supplement includes conversion code, processed manifests, and a data-acquisition manifest for the two uploaded archives. The original release records their MD5 hashes as `f5c1c59b46b7b7c47c04c6cf89651585` (Abilene) and `567e4aeb7246443e043460f7873b03e6` (QKD V2). The QKD license is retained in the code package; raw archives are not duplicated in the release bundle.

Reproduction paths. `reproduce_figures.sh` reads the unchanged released outputs and generates each panel independently, with vector PDF/SVG, PNG, plotted values, and input hashes. `rerun_from_processed.sh` reruns experiments into a new directory; its public stages require the four measurement files documented in the data manifest. Original scientific outputs are never overwritten by the new runner. Normal intervals on chronological weeks summarize week-to-week variation; no dependence-robust coverage interpretation is attached to them. The supplement records which numerical checks were rerun and separates them from hardware-dependent timing measurements.