Safe & Stable Dynamic Pricing Under Demand Drift: Offline Pretraining, Change-Point Detection, and Hard-Constraint Learning Guarantees

Liz Lemma Future Detective

January 16, 2026

Abstract

Dynamic pricing papers (including the source material's Q-learning simulation for retail pricing) typically optimize profit under a fixed demand model and do not address the 2026 deployment bottleneck: pricing systems must remain compliant under demand drift, regulatory price caps, and stability/anti-gouging rules. We propose a clean, tractable framework for deployable dynamic pricing: (i) offline pretraining from historical logs, (ii) online change-point detection for piecewise-stationary demand shifts, and (iii) a hard-constraint learning policy that enforces price floors/ceilings and bounded price changes at every period (auditable 'no-violation' guarantees). In a single-product setting with contextual demand and sub-Gaussian noise, we provide a high-probability regret bound that decomposes into within-regime learning error plus change-point detection/reset cost, scaling with the number and magnitude of regime shifts rather than horizon length. For linear (elasticity-style) demand, we additionally give closed-form characterizations of the constrained profit-maximizing price and the optimal adjustment path under stability constraints. Empirically, we outline an offline-to-online evaluation protocol on retailer logs and stress tests with synthetic regime shifts to quantify both revenue gains and compliance robustness.

Table of Contents

1. 1. Introduction: deployable dynamic pricing in 2026 (nonstationarity + governance constraints); contrast with static OR and unconstrained RL (link to source paper's Q-learning framing).

2. 2. Institutional constraints as primitives: price caps/floors, anti-gouging, bounded price volatility, and (optional) groupwise constraints; how these map to hard vs soft constraints.

3. 3. Model: piecewise-stationary contextual demand; profit; feasible action sets induced by caps and stability; benchmark (regime-wise best feasible policy).

4. 4. A tractable baseline: linear/elasticity demand; closed-form unconstrained optimum, constrained clipping, and optimal "move-to-target" path under stability constraint.

5. 5. Algorithm: Offline-to-online Safe Pricing (OSP): offline prior/confidence set + online optimistic/conservative choice + projection onto feasible action set; change-point detection and safe reset.

6. 6. Theory I (Safety): hard constraint satisfaction for all periods; extension to additional hard constraints.

7. 7. Theory II (Performance): dynamic regret decomposition (within-regime learning + detection delay); explicit dependence on regime count, noise, and drift magnitude.

8. 8. Practical evaluation: off-policy evaluation with guardrails, counterfactual bounds, and stress tests with simulated regime shifts; reporting compliance metrics (violations = 0 by design) and revenue metrics.

9. 9. Extensions: multi-SKU coupling, inventory/stockouts, CVaR risk constraints, and limited personalization with groupwise constraints (flagged as requiring numerical methods).

10. 10. Conclusion: implications for revenue management, auditing, and policy; limitations and next steps.

1 Introduction: deployable dynamic pricing in 2026

Dynamic pricing has moved from a specialized operations tool to a general-purpose decision layer embedded in consumer-facing platforms. In 2026, retailers and marketplaces routinely adjust posted prices in response to traffic, inventory conditions, competitor moves, and supply shocks; the same logic appears in delivery fees, ride-hailing multipliers, cloud-compute spot markets, and subscription upgrades. This ubiquity has changed the engineering problem. The central question is no longer whether a firm *can* compute a revenue-improving price from data, but whether it can do so in a way that is operationally stable, legally defensible, and robust to the nonstationarity that is now the rule rather than the exception.

Two empirical facts motivate our modeling choices. First, demand relationships drift and jump. A promotion by a competitor, a sudden supply disruption, a viral social-media mention, or a policy change by a platform can alter the mapping from price to demand within days. Treating demand as stationary over long horizons can therefore yield policies that look optimal on historical data yet systematically underperform in deployment. Second, pricing is governed. Caps, floors, anti-gouging rules, and internal compliance constraints constrain what a pricing system is allowed to post, and these restrictions are often *hard* in the sense that violations are unacceptable even if rare. In many organizations, the appropriate question is not "what price maximizes expected profit," but "what price maximizes expected profit among those that can be justified and audited."

Classical operations research provides an important baseline, and also illustrates the gap. In revenue management and price optimization, one typically posits a parametric demand model, estimates it offline, and computes an optimal price (or a policy) under stationarity. These methods deliver sharp prescriptions when the model is correct and the environment is stable, and they remain the default in many settings because they are interpretable and easy to govern. But they struggle when the demand curve changes midstream, because the data used to estimate the model mix multiple regimes. In such environments, a stationary optimizer is not merely "somewhat suboptimal"; it can be directionally wrong, continuing to push prices up when elasticity has increased, or holding prices down after willingness-to-pay has shifted upward. A deployable system must therefore learn online and also decide *when* what it has learned is no longer applicable.

At the other extreme, the reinforcement learning literature has made dynamic pricing a canonical example of sequential decision-making under uncertainty. In its most general form, the retailer is modeled as an agent interacting with an environment, receiving rewards (profits) and updating a value function. In particular, Q-learning and related temporal-difference methods have been used to learn pricing policies without specifying a demand model. This perspective is attractive: it promises to absorb rich state

variables and complex dynamics, and it aligns with the modern software stack in which pricing is one component of a broader decision system. Yet the same generality creates friction with governance. Standard RL relies on exploration that can be hard to justify ex ante, and its constraint handling is often indirect (e.g., adding penalties to rewards), which is poorly aligned with settings where regulators, platforms, or internal risk committees require *zero* violations of explicit price bounds or volatility limits. Moreover, generic RL methods are typically analyzed under stationarity assumptions; when the environment changes, they can become slow to adapt unless one adds additional machinery for resetting, discounting, or change detection.

Our objective is to articulate a middle ground: a model that is rich enough to capture the two facts above—nonstationarity and governance—while remaining structured enough to yield clear performance guarantees and an implementable algorithm. The central economic tradeoff is straightforward. The retailer wants to respond quickly when the demand regime changes, but must do so through a policy that is stable and defensible. Rapid adjustment creates customer and regulatory scrutiny; sluggish adjustment leaves profit on the table and can also generate distortions (e.g., stockouts or persistent mispricing). The model we develop makes this tension explicit by placing hard constraints directly on posted prices and by allowing demand to be *piecewise stationary*, so that learning is meaningful within regimes but must restart (or be discounted) across regime changes.

A key modeling decision is how to represent nonstationarity without losing analytical traction. Fully adversarial drift is a useful worst-case benchmark, but it is often too pessimistic for the environments we have in mind, where demand is stable for stretches and then shifts due to identifiable events. Conversely, assuming gradual drift only can miss the operational reality of sudden shocks. Piecewise stationarity captures a pragmatic middle: within each segment, the demand function is stable enough that the data are informative, while across segments, the system must detect that the mapping from price to demand has changed. This abstraction aligns with how practitioners talk about pricing systems: there are "normal times" punctuated by "event weeks" and "post-change" periods, each requiring distinct calibration.

Equally important is how we treat constraints. In deployment, the relevant constraints are typically written as simple, auditable rules: a price must lie between a context-dependent floor and cap; and it must not change by more than a prescribed increment from one period to the next. These rules can be motivated by regulation (anti-gouging statutes, sector-specific price controls), by platform governance (marketplace policies that restrict sudden price spikes), or by internal policy (brand protection, customer trust, and call-center load). Such constraints do not disappear when the model is uncertain; if anything, uncertainty strengthens the case for enforcing them mechanically. This is why we place them as primitives rather than as soft penalties: the algorithm must respect them period by period, not merely on

average.

With these primitives in place, we evaluate performance through *dynamic regret* relative to a natural benchmark: the best feasible stationary pricing rule within each regime. This benchmark is deliberately modest. It does not assume an oracle that can anticipate regime changes, nor does it allow instantaneous jumps that would themselves violate stability constraints. Instead, it asks whether an online system can track, with bounded loss, the best policy one could have run if one had known the regime in advance and had to obey the same governance rules. This framing makes the economics transparent: the unavoidable losses come from two sources, learning within a regime and paying a boundary cost around change points when the past stops being predictive. In turn, it makes clear what a deployable algorithm must do: learn quickly when the world is stable, and reset quickly (but not too often) when the world changes.

Our approach also clarifies what we do *not* attempt. We do not model strategic consumer behavior, collusion concerns among competing algorithms, or the full generality of multi-product substitution; these are important, but they complicate the inference and the constraint interface in ways that obscure the basic governance–adaptation tradeoff. We also do not claim that piecewise stationarity is the only relevant form of nonstationarity; gradual drift and seasonal cycling can be incorporated through context variables and alternative detection rules, but the core lesson remains: deployment requires separating "learning the demand curve" from "deciding whether the curve has changed."

The contribution of the framework is therefore conceptual as much as technical. By treating institutional constraints as hard primitives and nonstationarity as structured, we obtain an online pricing problem that is simultaneously realistic and analyzable. The rest of the paper builds this logic systematically: we formalize the governance constraints in a way that is auditable and implementable, we specify a learning-and-detection architecture that respects these constraints by construction, and we derive regret bounds that scale with the frequency and detectability of regime shifts rather than mechanically with the horizon. In doing so, we aim to illuminate how modern dynamic pricing systems can be both adaptive and governable—a requirement that, in 2026, is increasingly the difference between an algorithm that is profitable in simulation and one that can be safely deployed.

2 Institutional constraints as primitives: governable prices before optimal prices

In practice, a pricing system is rarely judged solely by the expected profit it delivers in a clean backtest. It is judged by whether it produces *defensible* and *operationally safe* prices period by period. This distinction matters

because many of the constraints that shape deployment are not naturally represented as statistical regularizers or "soft" preferences. They are written as rules, monitored by compliance teams, and enforced by platforms and regulators with little tolerance for exceptions. Our modeling choice in this paper is therefore to treat these institutional constraints as *primitives* of the pricing problem rather than as after-the-fact modifications to an unconstrained learning objective.

The most common constraints take the form of simple inequalities. A posted price must lie within a permissible range, and it must not move too abruptly. These restrictions appear across sectors: consumer staples during emergencies, marketplace listings subject to "price gouging" policies, and subscription products with internal brand-protection guidelines. The form is similar even when the motivation differs. A regulator may interpret extreme increases as exploitative; a platform may interpret them as harmful to user trust; a retailer may interpret them as generating call-center load, churn, or reputational damage. The common feature is that the organization can articulate the constraint ex ante and can audit it ex post.

Caps and floors as context-dependent rules. Price caps and floors are often *state contingent*. For instance, a grocery retailer may cap prices during a declared emergency, while allowing wider latitude in normal periods; a marketplace may impose tighter caps in categories with a history of abuse; or an internal policy may require a minimum margin, effectively imposing a floor tied to marginal cost. These practices motivate representing the allowable interval as functions of observed context,

$$p \in [\underline{p}(x), \overline{p}(x)],$$

where x collects the variables that compliance teams can point to when justifying why a particular restriction applied (location, category, emergency flags, procurement conditions, or policy regime indicators supplied by the platform). Importantly, we do not assume these bounds are "economically optimal"; rather, they are *institutional* objects: they summarize what is permissible, not what is best.

This perspective also clarifies how cost enters governance. Many anti-gouging statutes and internal rules are written in terms of markups. A simplified representation is

$$p \leq c(x)(1 + m(x)),$$

for some permitted markup $m(x)$. This is still a cap, but one that shifts with observed costs. Modeling caps and floors as functions of context allows us to incorporate such cost-based policies without endogenizing the policy itself.

Anti-gouging as a hard constraint rather than a moral preference. Anti-gouging is often discussed as a fairness or welfare issue, but in deployment it functions as a *compliance constraint* whose violation is catastrophic. From an algorithm-design standpoint, this pushes us away from penalty-based approaches. A soft penalty of the form "subtract $\lambda \cdot \mathbf{1}\{p \text{ too high}\}$" does not guarantee compliance under exploration noise, estimation error, or distribution shift. By contrast, a hard cap does: it mechanically excludes the forbidden region from the action space.

Moreover, anti-gouging policies are frequently defined relative to a *reference price* (e.g., "no more than $g\%$ above the pre-emergency price"). One can encode this in several equivalent ways. If $p^{\mathrm{ref}}(x)$ denotes a context-dependent reference, then a rule like

$$p_t \leq (1+g)\, p^{\mathrm{ref}}(x_t)$$

is simply a particular choice of $\overline{p}(x_t)$. Alternatively, if the rule is operationalized as "do not raise prices faster than a certain rate during emergencies," it can be expressed through a stability constraint (discussed next). The key modeling point is that these are not nuances to be handled by an objective function; they are constraints that define what actions are admissible.

Bounded price volatility as a stability and auditability requirement. Even when caps and floors are generous, organizations often impose limits on how quickly posted prices can change. The economic motivations are varied: customers perceive volatile prices as unfair; sudden jumps attract scrutiny; and rapid reversals create operational chaos (e.g., price-matching disputes, returns, and manual overrides). Technically, bounded volatility is also a way to ensure that exploration remains controlled and that any misestimation does not translate into extreme realized outcomes.

We represent such requirements in a particularly auditable form: a bound on per-period movement,

$$|p_t - p_{t-1}| \leq \Delta.$$

This rule is attractive precisely because it is simple: it does not require specifying a demand model or estimating consumer surplus to decide whether a price is "reasonable." It only requires comparing today's price to yesterday's, which makes it easy to implement and to monitor. In many organizations, this simplicity is not a weakness; it is the reason the rule survives legal review and cross-functional governance.

Stability constraints also interact with anti-gouging in a natural way. When an emergency begins, a cap may suddenly tighten; when it ends, it may relax. A stability rule ensures that the transition into the new permissible region occurs gradually, which reduces the risk that the algorithm whipsaws prices in response to noisy demand signals or misclassified contexts. In other

words, stability is not only a consumer-facing policy; it is a control-theoretic safety device.

Optional groupwise constraints and their operational meaning. A third family of restrictions arises from concerns about discrimination, fairness, or parity across segments. Even in single-product settings, firms may constrain prices across geography, user type, or acquisition channel. Sometimes the rule is strict parity ("the same posted price for all customers"), but more commonly it is a bounded disparity ("prices may differ, but not by more than η across protected groups"), or a constraint on the mapping from observable contexts to prices (e.g., "do not use certain sensitive attributes directly").

In a single-price-per-period environment, many groupwise constraints can be represented by controlling what information enters x and by specifying segment-dependent bounds. For example, if x includes a group label g, then a parity requirement can be encoded by forcing $\underline{p}(x)$ and $\overline{p}(x)$ to coincide across values of g, effectively removing group-conditional variation from the feasible set. More complex parity requirements couple decisions across contexts (e.g., a constraint that two segments' prices must remain within η of each other at the same time). Such coupled constraints quickly move beyond the interval action sets we focus on here, and they become closer to multi-action feasibility problems. We view them as important extensions, but we keep the core model centered on constraints that can be audited period by period from a single posted price.

Hard versus soft constraints: why we insist on mechanical feasibility. It is tempting to treat governance requirements as additional terms in the objective, because doing so preserves the form of a standard learning problem. But this move conflates two distinct questions: (i) what the organization prefers, and (ii) what the organization is *allowed* to do. Preferences can be balanced; permissions cannot. A "soft" anti-gouging penalty can be outweighed by an estimated profit gain in a regime where the model (wrongly) believes demand is inelastic. A hard cap cannot.

For this reason, we distinguish sharply between *hard* constraints that must hold pathwise (every t) and *soft* considerations that can be traded off in expectation. In our framework, caps, floors, and stability limits are hard. They define the feasible set of actions, and any learning algorithm must output a price inside that set. Soft considerations—such as preferring smoother prices than the maximum allowed, or preferring prices that are "close" to a reference level even when not required—can still be incorporated, but they should appear as secondary design choices (e.g., in the selection of a candidate price before enforcing feasibility), not as substitutes for feasibility itself.

What this buys us, and what it costs. Treating institutional constraints as primitives yields two benefits. First, it separates compliance from inference: the system can explore and learn about demand without risking forbidden outputs, because feasibility is enforced mechanically. Second, it makes guarantees meaningful: a regret bound is only operationally relevant if the algorithm it describes can actually be deployed under the same rules that govern human pricing.

The cost is that hard constraints can be conservative. They may preclude profitable responses to genuine shocks, especially when caps are tight or when Δ is small. From an economic perspective, this conservatism is not a bug but a feature: it is the model's way of representing the shadow cost of governance. In the next section, we formalize these constraints as an induced feasible action set that depends on context and on last period's price, and we evaluate learning performance relative to a benchmark that is itself subject to the same institutional limits. This keeps the comparison honest: we do not credit an algorithm for profits that could only be attained by violating the rules under which it must operate.

3 Model: piecewise-stationary contextual demand under hard feasibility

We now formalize the pricing environment implied by the institutional primitives described above. The central modeling choice is that learning takes place *inside* an action space defined by auditable constraints, and nonstationarity enters through discrete, unobserved shifts in the demand law. This lets us separate three objects that are often conflated in practice: (i) what is *permitted* (caps, floors, and stability), (ii) what is *known* (marginal cost and observed context), and (iii) what must be *learned and tracked* (the demand response, which can change over time).

Horizon, observables, and timing. Time is indexed by $t \in \{1, \dots, T\}$. At the start of each period, the retailer observes a context vector $x_t \in \mathcal{X}$ summarizing cost shifters and demand-relevant state (seasonality, traffic, procurement conditions, policy flags, etc.). After observing x_t, the retailer posts a single price p_t. Demand then realizes as a scalar quantity y_t (units sold), and profit is computed. Formally, the within-period timing is:

1. observe x_t,
2. choose p_t,
3. observe y_t,
4. receive profit π_t and update the pricing rule.

The retailer observes the history $(x_s, p_s, y_s)_{s \le t}$ but does not observe the latent demand regime or its parameters.

Hard constraints induce a state-dependent feasible set. Institutional restrictions enter as hard constraints on posted prices. We allow both (i) a context-dependent permissible interval $[\underline{p}(x_t), \overline{p}(x_t)]$ and (ii) a stability constraint limiting per-period movement. Given last period's posted price p_{t-1} and current context x_t, the feasible action set is the closed interval

$$\mathcal{A}_t(x_t, p_{t-1}) = \Big[\max\{\underline{p}(x_t),\, p_{t-1} - \Delta\},\ \min\{\overline{p}(x_t),\, p_{t-1} + \Delta\}\Big],$$

where $\Delta \ge 0$ is a fixed stability parameter and p_0 is given. Two properties are worth emphasizing. First, feasibility is *path dependent*: the stability constraint couples decisions across time, so the set of admissible prices today depends on what we posted yesterday. Second, feasibility is *auditable*: $\underline{p}(\cdot)$, $\overline{p}(\cdot)$, and Δ are known objects, so whether a price violated policy can be verified without reference to any demand model.

In implementation, we will often describe an algorithm as proposing a candidate price $\tilde{p}_t \in \mathbb{R}$ (e.g., an optimistic or greedy estimate) and then mechanically enforcing compliance via projection,

$$p_t = \Pi_{\mathcal{A}_t(x_t, p_{t-1})}(\tilde{p}_t).$$

This creates a clean separation between statistical learning (which may err) and compliance (which must not).

Demand with latent regimes and contextual dependence. Demand responds to price and context but may shift over time in a way that is not directly observed. We model this through a piecewise-stationary regime index $r_t \in \{1, \dots, R\}$ with change points

$$1 = \tau_1 < \tau_2 < \cdots < \tau_R \le T, \qquad r_t = r \ \text{ for } t \in \mathcal{I}_r := [\tau_r, \tau_{r+1} - 1],$$

where $\tau_{R+1} = T + 1$ and regime length is $L_r := \tau_{r+1} - \tau_r$. In regime r, expected demand is given by an unknown function $D_r(p, x)$:

$$\mathbb{E}[y_t \mid p_t, x_t, r_t = r] = D_r(p_t, x_t).$$

To allow high-probability learning guarantees while remaining agnostic about the exact noise law, we assume the demand shock is conditionally σ-sub-Gaussian. Equivalently, we can write

$$y_t = D_{r_t}(p_t, x_t) + \varepsilon_t, \qquad \mathbb{E}[\varepsilon_t \mid \mathcal{F}_{t-1}, x_t, p_t] = 0,$$

and for all $\lambda \in \mathbb{R}$,

$$\mathbb{E}\big[\exp(\lambda \varepsilon_t) \mid \mathcal{F}_{t-1}, x_t, p_t\big] \le \exp\left(\tfrac{\lambda^2 \sigma^2}{2}\right),$$

where $\mathcal{F}_{t-1}$ denotes the sigma-field generated by past observations. This assumption is standard in bandit-style analyses: it is strong enough to obtain concentration yet weak enough to include many bounded or light-tailed demand disturbances.

Costs and profits. We treat marginal cost as known given context, $c(x_t)$. This captures the practical reality that many retailers can observe (or at least forecast with low error) procurement costs and fees, whereas demand is the harder object to infer. Profit is realized as

$$\pi_t = (p_t - c(x_t))\, y_t,$$

and the regime-r conditional expected profit at context x and price p is

$$g_r(p, x) := \mathbb{E}[\pi_t \mid p_t = p, x_t = x, r_t = r] = (p - c(x))\, D_r(p, x).$$

We do not impose a particular functional form on D_r in the general model; the tractable linear specification appears in the next section.

What we compete against: an honest constrained benchmark. Because feasibility restrictions are non-negotiable, the relevant notion of performance is regret relative to the best policy that *also* respects the same hard constraints. We adopt a regime-wise benchmark that is stationary within each regime in the sense of using a time-invariant pricing rule for that regime, but still respects the path constraint induced by Δ.

Concretely, fix a regime r. Consider a benchmark that knows D_r (but not future regimes) and, within $\mathcal{I}_r$, selects each period a feasible price maximizing expected profit subject to the same caps/floors and stability constraint along its own path. Let p^*_{t-1} denote the benchmark's previous price. The benchmark's one-step optimal expected profit is

$$\pi^*_t(r) := \max_{p \in \mathcal{A}_t(x_t, p^*_{t-1})} (p - c(x_t))\, D_r(p, x_t).$$

This definition is intentionally "institutional": it allows the benchmark to fully exploit regime-r demand knowledge, but it forbids it from making an infeasible jump or posting an impermissible price. In particular, when Δ is small, even an informed benchmark may need several periods to move toward its preferred level, so it is inappropriate to compare an online algorithm to an unconstrained clairvoyant that can jump instantly.

Dynamic regret under piecewise stationarity. We measure learning-and-tracking performance through dynamic regret relative to the regime-wise benchmark:

$$\mathrm{Reg}_T := \sum_{t=1}^{T} \Big(\mathbb{E}[\pi^*_t(r_t)] - \mathbb{E}[\pi_t] \Big).$$

Two aspects are doing work here. First, the comparator is *dynamic* because the regime index r_t changes over time; we do not ask the learner to compete with a single stationary policy over $t = 1, \ldots, T$ when the environment itself is changing. Second, the comparator is *feasible* because it internalizes the same action constraints as the learner. This keeps the comparison aligned with deployment: the regret quantity answers the operational question of how much profit is lost due to learning and delayed adaptation, *holding fixed* the organization's governance rules.

Our objective in the remainder of the paper is to exhibit a pricing algorithm that (i) satisfies $p_t \in \mathcal{A}_t(x_t, p_{t-1})$ for all t by construction, and (ii) achieves regret that scales with the difficulty of learning within regimes and with the frequency and detectability of regime changes, rather than scaling linearly with T. The next section illustrates these ideas in a tractable baseline where D_r is linear in price, making the constrained optimum and the stability-constrained adjustment path fully explicit.

4 A tractable baseline: linear and elasticity demand with explicit constrained optima

To build intuition for both the benchmarking notion and the algorithmic design that follows, it is useful to study a demand class in which (i) the profit maximizer has a closed form and (ii) the effect of hard constraints is transparent. The point is not that real demand is literally linear, but that the linear and constant-elasticity specifications deliver interpretable "targets" and make clear how caps/floors and stability constraints transform a static pricing problem into a path problem.

Linear demand within a regime. Fix a regime r and consider the specification

$$D_r(p, x) = a_r(x) - b_r(x)\, p, \qquad b_r(x) > 0,$$

with the understood truncation $D_r(p, x) := \max\{0,\, a_r(x) - b_r(x)p\}$ when we want to rule out negative quantities. Here $a_r(x)$ is a context-dependent intercept (baseline demand) and $b_r(x)$ is a context-dependent slope (price sensitivity). When x includes seasonality and marketing variables, allowing $a_r(x)$ and $b_r(x)$ to vary with x captures the practical reality that both market size and willingness-to-pay shift predictably with observed conditions, even absent a regime change.

Given known cost $c(x)$, the regime-r conditional expected profit is the quadratic

$$g_r(p, x) = (p - c(x))\,\big(a_r(x) - b_r(x)p\big) = -b_r(x)p^2 + \big(a_r(x) + b_r(x)c(x)\big)p - a_r(x)c(x),$$

ignoring truncation for the moment. Concavity is immediate: for each fixed (r, x) we have $\partial_{pp} g_r(p, x) = -2b_r(x) < 0$. This strict curvature will be the key reason constraints "clip" rather than create multiple local optima.

Unconstrained optimum and its economic meaning. Differentiating and setting the first-order condition to zero gives

$$\partial_p g_r(p, x) = a_r(x) - 2b_r(x)p + b_r(x)c(x) = 0,$$

so the unique unconstrained maximizer is

$$p_r^*(x) = \frac{a_r(x) + b_r(x)c(x)}{2b_r(x)} = \frac{1}{2}\Big(\frac{a_r(x)}{b_r(x)} + c(x)\Big).$$

The decomposition is instructive: $\frac{a_r(x)}{b_r(x)}$ is the choke price (the price at which the untruncated linear demand would hit zero), and the optimal unconstrained price is the midpoint between the choke price and marginal cost. When the demand intercept rises (higher $a_r(x)$), the choke price increases and so does $p_r^*(x)$. When the slope rises (higher $b_r(x)$, more price sensitivity), both the choke price and the markup component shrink, pushing the optimal price down.

If we impose truncation explicitly, then posting $p \geq a_r(x)/b_r(x)$ yields zero expected quantity, so such prices are weakly dominated by any feasible price that generates strictly positive demand. Operationally, truncation can therefore be viewed as an *endogenous cap* on revenue-relevant prices; in constrained problems it is convenient to treat $a_r(x)/b_r(x)$ as an additional (soft) upper bound that the optimal policy will not want to cross.

Caps and floors: constrained optimum as clipping. Now impose only the context-dependent cap/floor constraint $p \in [\underline{p}(x), \overline{p}(x)]$, abstracting from stability for a moment. Because $g_r(\cdot, x)$ is concave, the constrained maximizer is obtained by Euclidean projection ("clipping") of the unconstrained optimizer onto the interval:

$$p_r^{\text{clip}}(x) = \min\Big\{\overline{p}(x),\, \max\{\underline{p}(x),\, p_r^*(x)\}\Big\}.$$

With truncation, one may further clip at the choke price, replacing $\overline{p}(x)$ by $\min\{\overline{p}(x),\, a_r(x)/b_r(x)\}$, but in many applications the institutional cap is already below any economically relevant choke price. The key point is that the policy objects $\underline{p}(\cdot)$ and $\overline{p}(\cdot)$ enter *mechanically*: once (a_r, b_r) are known, the constrained optimum is immediate and auditable.

Stability transforms a static target into a path problem. The stability constraint $|p_t - p_{t-1}| \leq \Delta$ matters even if (r, x) are fixed, because

it prevents instantaneous movement to $p_r^{\text{clip}}(x)$. In the simplest stationary case—a fixed regime r and constant context $x_t \equiv x$—the retailer faces the problem of choosing a feasible price *path* to maximize $\sum_t g_r(p_t, x)$ subject to the per-period movement constraint and the cap/floor bounds.

With concave per-period profit and identical primitives over time, there is a particularly sharp characterization. Let $p^{\text{tar}} := p_r^{\text{clip}}(x)$ denote the (box-)constrained target price. Then any delay in approaching p^{tar} sacrifices profit each period without creating offsetting gains later, since there is no intertemporal coupling other than the movement constraint itself. Consequently, the optimal feasible path moves toward p^{tar} as fast as allowed, and stops once it reaches the target (or the relevant boundary). Formally, starting from p_0, the path recursion is

$$p_t^{\text{path}} = \Pi_{[\underline{p}(x),\,\overline{p}(x)]}\Big(p_{t-1}^{\text{path}} + \operatorname{sgn}\big(p^{\text{tar}} - p_{t-1}^{\text{path}}\big) \cdot \min\{\Delta,\ |p^{\text{tar}} - p_{t-1}^{\text{path}}|\}\Big).$$

This "move-to-target" rule makes the welfare role of Δ transparent: smaller Δ does not change the long-run target but slows the transition, creating transient losses after shocks to costs, demand, or policy bounds.

Within-regime variation in context: a sequence of targets. When x_t varies over time within a regime, there is no single stationary target; instead, each period has its own myopic optimizer $p_r^*(x_t)$ and clipped target $p_r^{\text{clip}}(x_t)$. Even then, the linear model still yields a convenient one-step structure: given a previous posted price p_{t-1}, the per-period constrained maximizer (holding r fixed) solves

$$\max_{p \in \mathcal{A}_t(x_t, p_{t-1})} g_r(p, x_t),$$

and by concavity this is again obtained by projecting the unconstrained optimizer $p_r^*(x_t)$ onto the *state-dependent* feasible set $\mathcal{A}_t(x_t, p_{t-1})$. Thus, in the linear case, the stability constraint can be interpreted as forcing a bounded-speed tracking problem in which the desired level changes with x_t (and, in our larger model, also with r_t).

Constant-elasticity demand as an alternative benchmark. A second workhorse specification replaces linearity with multiplicative scale and curvature:

$$D_r(p, x) = A_r(x)\, p^{-\eta_r(x)}, \qquad A_r(x) > 0,\ \eta_r(x) > 0,$$

again with profit $g_r(p, x) = (p - c(x))D_r(p, x)$. When $\eta_r(x) > 1$ and $c(x)$ is locally constant in p, the interior first-order condition $D_r(p, x) + (p - c(x))\partial_p D_r(p, x) = 0$ yields the familiar markup rule

$$\frac{p - c(x)}{p} = \frac{1}{\eta_r(x)}, \qquad \text{so} \qquad p_r^*(x) = \frac{\eta_r(x)}{\eta_r(x) - 1}\, c(x).$$

This representation is attractive in settings where positivity and proportional responses are important, and it makes explicit that higher elasticity (larger $\eta_r(x)$) implies a lower optimal markup. Under caps/floors, the constrained optimum is again a clipping of $p_r^*(x)$, and under stability one again obtains a bounded-speed adjustment toward the constrained target whenever per-period profit is concave in the relevant range. Unlike the linear case, however, concavity may hold only locally, so we view elasticity demand mainly as a robustness check on the qualitative lessons rather than as the main vehicle for closed-form regret analysis.

Why this baseline matters for the algorithm. The linear (and, to a lesser extent, elasticity) models clarify the separation we exploit in the next section: statistical learning delivers an estimate (or confidence region) for primitives such as $a_r(x)$ and $b_r(x)$, which induces a candidate "target" price (e.g., $p_r^*(x)$ or an optimistic variant), while compliance is enforced by projecting that candidate into the hard feasible set. Moreover, because linear demand is a regression model in price, it naturally supports least-squares estimation, residual diagnostics, and change-point tests—tools that will underpin our Offline-to-online Safe Pricing procedure.

5 Algorithm: Offline-to-online Safe Pricing (OSP)

Our objective is to combine two requirements that are often treated separately in practice: (i) *statistical adaptivity* to unknown and shifting demand, and (ii) *mechanical compliance* with hard pricing constraints that must hold period by period. The Offline-to-online Safe Pricing (OSP) procedure is built as a modular stack. An *estimation layer* produces a probabilistic description of demand primitives; a *decision layer* converts that description into a candidate (possibly unconstrained) price; a *safety layer* enforces all hard constraints by projection; and a *monitoring layer* runs change-point detection and triggers resets when the regime appears to have shifted.

Offline initialization: a prior confidence set rather than a point estimate. We begin with an offline dataset $\mathcal{D}_0 = \{(x_i, p_i, y_i)\}_{i=1}^{n_0}$ drawn from historical operation (or a pilot). The role of $\mathcal{D}_0$ is not to pin down the demand function exactly, but to provide a *calibrated uncertainty set* that can be carried into the online phase. Concretely, we posit a parametric demand family $\{D_\theta(p, x) : \theta \in \Theta\}$ within a regime (e.g., linear demand with $\theta = (a, b)$). We compute an offline estimator $\hat{\theta}_0$ and an associated design matrix V_0 (e.g., the ridge-regularized Gram matrix). From standard self-normalized concentration, we can form an elliptical confidence region

$$\mathcal{C}_0(\delta) = \Big\{\theta \in \Theta : \ \|\theta - \hat{\theta}_0\|_{V_0} \le \beta_0(\delta)\Big\},$$

chosen so that θ lies in $\mathcal{C}_0(\delta)$ with probability at least $1-\delta$ under the offline data-generating process. Two implementation details matter. First, we treat $\mathcal{C}_0(\delta)$ as an *input object* that may be produced by any compliant estimation pipeline (including covariate selection and robust regression), as long as it yields a valid high-probability region. Second, because the online algorithm will occasionally reset, $\mathcal{C}_0(\delta)$ functions as a reusable "prior" that prevents each post-reset learning phase from starting from scratch.

Online learning within a regime: optimistic targets with optional conservatism. During a regime segment, OSP maintains an online confidence set $\mathcal{C}_t(\delta)$ updated from the data observed since the most recent reset (optionally combined with the offline prior through regularization). The decision layer then assigns each feasible price a plausible profit level by optimizing over demand parameters consistent with the data. A canonical choice is an optimistic (UCB-style) criterion:

$$U_t(p) := \max_{\theta \in \mathcal{C}_t(\delta)} (p - c(x_t))\, D_\theta(p, x_t), \qquad \tilde{p}_t \in \arg\max_{p \in \mathbb{R}} U_t(p),$$

where $\tilde{p}_t$ is a *candidate* price that need not respect caps, floors, or stability. In the linear-demand case, $U_t(p)$ is typically a concave quadratic envelope in p (after optimizing over θ in the ellipsoid), so $\tilde{p}_t$ can be computed quickly and deterministically.

In applications where managers or regulators prefer a more conservative posture, we can temper optimism by mixing in a lower-confidence proxy. For example, define a pessimistic (LCB-style) value

$$L_t(p) := \min_{\theta \in \mathcal{C}_t(\delta)} (p - c(x_t))\, D_\theta(p, x_t),$$

and choose $\tilde{p}_t$ to maximize a convex combination $\lambda U_t(p) + (1-\lambda) L_t(p)$ with $\lambda \in [0,1]$, or impose explicit guardrails such as $(p - c(x_t))\widehat{D}_t(p, x_t) \geq 0$ to avoid intentionally pricing into predicted negative margins when forecasts are unreliable.[1] The key separation is that *the statistical criterion generates only a recommendation*; the constraint system remains binding at implementation time.

Safety layer: projection as an auditable compliance mechanism. Given x_t and the last posted price p_{t-1}, the feasible action set is the interval

$$\mathcal{A}_t(x_t, p_{t-1}) = \Big[\max\{\underline{p}(x_t),\, p_{t-1} - \Delta\},\ \min\{\overline{p}(x_t),\, p_{t-1} + \Delta\}\Big].$$

OSP *always* implements the projected action

$$p_t = \Pi_{\mathcal{A}_t(x_t, p_{t-1})}(\tilde{p}_t).$$

[1] These optional conservatism features change constants and may affect regret rates, but they do not affect the mechanical safety guarantees that follow from projection.

This step is deliberately "dumb": it does not depend on demand estimates, on whether the learning model is misspecified, or on whether the change-point detector is behaving well. In operational terms, projection is an easily logged and audited transformation from a candidate price to a compliant price, yielding a clean separation between (i) a potentially complex recommendation engine and (ii) a simple compliance wrapper that guarantees $p_t \in [\underline{p}(x_t), \overline{p}(x_t)]$ and $|p_t - p_{t-1}| \le \Delta$ deterministically.

Online updates: regression-style estimation and residual tracking. After posting p_t, we observe y_t and update the estimator and confidence set. In the linear case, with features $\phi_t = \phi(p_t, x_t)$ (e.g., $\phi_t = (1, -p_t)$ possibly interacted with x_t), we maintain regularized least squares

$$\hat{\theta}_t = \arg\min_{\theta \in \Theta} \sum_{s \in \mathcal{S}_t} \big(y_s - \langle \theta, \phi_s \rangle\big)^2 + \lambda \|\theta\|^2, \qquad V_t = \lambda I + \sum_{s \in \mathcal{S}_t} \phi_s \phi_s^\top,$$

where $\mathcal{S}_t$ indexes observations since the last reset (and optionally includes offline pseudo-observations encoding $\mathcal{C}_0$). The associated confidence radius $\beta_t(\delta)$ is updated using the usual σ-sub-Gaussian bounds. For monitoring, we also compute a one-step residual

$$e_t := y_t - \widehat{D}_t(p_t, x_t),$$

and (when needed) a standardized residual $z_t = e_t / \sqrt{1 + \|\phi_t\|^2_{V_t^{-1}}}$, which stabilizes variance across prices and contexts.

Change-point detection: residual-based alarms and safe resets. Nonstationarity enters through regime changes, so OSP continuously tests whether recent data remain consistent with the current within-regime model. A simple and effective choice is a windowed CUSUM/GLR-style detector applied to residuals or parameter estimates. For instance, for a window length w, define a statistic

$$S_t = \max_{k \in \{t-w, \dots, t-1\}} \left| \sum_{s=k+1}^{t} z_s \right|,$$

and trigger an alarm if $S_t > \gamma(w, \delta)$ for a threshold γ calibrated to control false alarms under the no-change null. More structured alternatives compare pre- and post-split estimates $\hat{\theta}_k$ and $\hat{\theta}_{k:t}$ via a norm $\|\hat{\theta}_{k:t} - \hat{\theta}_k\|$, which is natural when the primary shifts are in demand primitives rather than in noise.

When an alarm fires, we *reset the learning state*: we start a new segment, discard (or downweight) pre-change online data, and reinitialize the confidence set to the offline prior $\mathcal{C}_0(\delta)$ (or to a combination of $\mathcal{C}_0(\delta)$ and

a short post-alarm warm start). Importantly, a reset does *not* override the posted price history. The next period still uses the actual p_t as the anchor for the stability constraint, and projection continues to enforce $|p_{t+1} - p_t| \leq \Delta$. Thus, even if the post-reset "target" price implied by the new estimates jumps sharply, the implemented prices adjust only at the permitted speed.

Summary and interface to the safety theory. OSP can be read as a disciplined workflow: offline data yield a calibrated uncertainty set; online learning converts uncertainty into an optimistic (or tempered) candidate; projection enforces hard constraints exactly; and a residual-based detector limits the damage from regime changes by triggering safe resets. The next section formalizes the resulting safety guarantee: constraint satisfaction holds for all t by construction, and the same projection wrapper extends immediately to additional hard constraints that can be encoded as a closed feasible set.

6 Theory I (Safety): Hard Constraint Satisfaction for All Periods

The central design choice in OSP is that *safety is enforced at the action interface*, not in the statistical model. Concretely, the learning and detection components are allowed to be wrong (misspecification), slow (detection delay), or even adversarially perturbed (software bugs), yet the implemented price sequence must remain compliant period by period. This requirement is naturally pathwise: unlike statistical guarantees, a single violation may be unacceptable in regulated or reputationally sensitive settings. Our safety theory therefore treats the candidate price $\tilde{p}_t$ as an arbitrary real number and shows that the *projection wrapper* deterministically enforces all hard constraints.

Nonempty feasibility as an operational consistency condition. Because caps/floors and stability are imposed simultaneously, we require that the induced feasible action set is nonempty at each t:

$$\mathcal{A}_t(x_t, p_{t-1}) = \Big[\max\{\underline{p}(x_t),\, p_{t-1} - \Delta\},\ \min\{\overline{p}(x_t),\, p_{t-1} + \Delta\}\Big] \neq \emptyset.$$

A sufficient condition is $\underline{p}(x) \leq \overline{p}(x)$ for all x and that the platform does not tighten bounds so abruptly that the previous compliant price p_{t-1} becomes infeasible *and* cannot be moved into feasibility within step size Δ. In practice, when such conflicts can occur (e.g., context-dependent caps that change sharply with x_t), one typically specifies a priority rule (caps/floors override stability, or vice versa). Our analysis accommodates this by defining $\mathcal{A}_t$ as the set of actions permitted by the chosen priority rule; the results below then apply verbatim.

Safety by projection. Given any closed interval $I = [\ell, u]$ with $\ell \le u$, the Euclidean projection map is

$$\Pi_I(z) := \arg\min_{p \in I} (p - z)^2 = \min\{u, \max\{\ell, z\}\}.$$

OSP implements

$$p_t = \Pi_{\mathcal{A}_t(x_t, p_{t-1})}(\tilde{p}_t),$$

where $\tilde{p}_t$ is produced by the decision layer. The next proposition formalizes the pathwise compliance guarantee.

Proposition (hard-constraint safety by design). Fix any sequence of contexts $(x_t)_{t=1}^T$ and any sequence of candidate prices $(\tilde{p}_t)_{t=1}^T$ in $\mathbb{R}$. Suppose $\mathcal{A}_t(x_t, p_{t-1})$ is nonempty for each t and set $p_t = \Pi_{\mathcal{A}_t(x_t, p_{t-1})}(\tilde{p}_t)$. Then for all t,

$$p_t \in [\underline{p}(x_t), \overline{p}(x_t)] \qquad \text{and} \qquad |p_t - p_{t-1}| \le \Delta.$$

Proof (direct). By construction, $\mathcal{A}_t(x_t, p_{t-1})$ is the intersection of the cap/floor interval and the stability interval around p_{t-1}:

$$\mathcal{A}_t(x_t, p_{t-1}) = [\underline{p}(x_t), \overline{p}(x_t)] \ \cap \ [p_{t-1} - \Delta, p_{t-1} + \Delta].$$

Projection maps any real number into the set onto which we project, hence $p_t \in \mathcal{A}_t(x_t, p_{t-1})$. Membership in the intersection implies both $p_t \in [\underline{p}(x_t), \overline{p}(x_t)]$ and $p_t \in [p_{t-1} - \Delta, p_{t-1} + \Delta]$, the latter being equivalent to $|p_t - p_{t-1}| \le \Delta$. □

Interpretation: decoupling model risk from compliance risk. The proposition is intentionally stronger than an in-expectation statement: it holds for *every* realized path of $(x_t, \tilde{p}_t)$ and therefore for every realization of the stochastic demand process, every update of the estimator, and every behavior of the change-point detector. In operational terms, this decoupling is valuable because it converts compliance into a simple, loggable transformation. An auditor can reconstruct $\mathcal{A}_t$ from (x_t, p_{t-1}) and verify that the posted p_t equals $\Pi_{\mathcal{A}_t}(\tilde{p}_t)$, without inspecting the learning code or the demand model.

Extension 1: additional hard constraints as closed feasible sets. The same logic extends immediately beyond cap/floor and one-step stability. Let $\mathcal{F}_t$ be any (possibly context- and history-dependent) *hard feasibility set* such that the platform requires $p_t \in \mathcal{F}_t$ deterministically. If $\mathcal{F}_t \subseteq \mathbb{R}$ is closed and nonempty, we can define the implemented action by

$$p_t \in \Pi_{\mathcal{F}_t}(\tilde{p}_t) \in \arg\min_{p \in \mathcal{F}_t} (p - \tilde{p}_t)^2.$$

When $\mathcal{F}_t$ is a closed interval, $\Pi_{\mathcal{F}_t}$ reduces to clipping; when $\mathcal{F}_t$ is a finite set (e.g., prices must be in \$0.05 increments), $\Pi_{\mathcal{F}_t}$ is the nearest feasible price (ties can be broken deterministically). In all cases, the safety claim is tautological: $p_t \in \mathcal{F}_t$ holds by construction. Convexity is not required for feasibility, only for uniqueness and computational convenience.

Examples. (i) *Percentage-change stability:* replace $|p_t - p_{t-1}| \le \Delta$ with $p_t \in [(1-\eta)p_{t-1}, (1+\eta)p_{t-1}]$ for $\eta \in (0,1)$ and intersect with $[\underline{p}(x_t), \overline{p}(x_t)]$. This again yields a closed interval and clipping enforces compliance.

(ii) *Margin guardrails:* enforce $p_t \ge c(x_t) + m_{\min}$ for a required minimum unit margin $m_{\min} \ge 0$ (or, more generally, $p_t \in [c(x_t) + m_{\min}, c(x_t) + m_{\max}]$). This is another cap/floor with context-dependent bounds.

(iii) *Forbidden regions:* prohibit prices in an interval $(p_1^{\text{bad}}, p_2^{\text{bad}})$ (e.g., regulatory "no-go" bands). The feasible set becomes a union of closed intervals. Projection is still well defined as nearest-point selection and yields deterministic feasibility, though the selected price may jump between components; if jumps are also constrained, one encodes stability directly in $\mathcal{F}_t$.

Extension 2: multi-product pricing and general constraints. If the action is a price vector $p_t \in \mathbb{R}^m$ (multiple products or zones), hard constraints often include componentwise bounds, cross-product parity rules, and stability constraints of the form $\|p_t - p_{t-1}\| \le \Delta$ under a chosen norm. Let $\mathcal{F}_t \subseteq \mathbb{R}^m$ denote the resulting feasible polytope (or more general closed set). Implementing

$$p_t \in \arg\min_{p \in \mathcal{F}_t} \|p - \tilde{p}_t\|_2^2$$

again yields zero violations. When $\mathcal{F}_t$ is convex, the projection is unique and computable via a quadratic program, which is precisely the type of routine that platforms can harden and audit.

Limitations: what safety does *not* guarantee. Projection ensures compliance with encoded constraints, but it does not ensure that those constraints are themselves well chosen. For example, a tight stability bound Δ can mechanically prevent rapid adjustment after a regime change; conversely, a loose Δ may satisfy policy but raise anti-gouging concerns. Likewise, feasibility conflicts must be resolved by design (via nonempty $\mathcal{A}_t$ or an explicit priority rule). These are not statistical issues but governance choices. The role of the safety layer is narrower: given a set of hard requirements, it guarantees that the algorithm never violates them, independent of how demand is learned or how regimes are detected.

7 Theory II (Performance): Regret Under Piecewise Stationarity

Having separated compliance from the statistical layer, we now ask a different question: *how costly is learning when demand shifts over time?* Our performance objective is dynamic regret against a benchmark that is deliberately modest but operationally meaningful: in each regime, compare to the best *feasible stationary* pricing rule for that regime, recognizing that the benchmark itself must obey the same cap/floor and step-size constraints. This choice avoids an unrealistic comparator that can instantaneously jump to the new optimum after a change point, which would overstate what is achievable under stability constraints.

Benchmark and regret. Let $\mathcal{I}_r = [\tau_r, \tau_{r+1} - 1]$ denote regime r and write $r_t = r$ for $t \in \mathcal{I}_r$. For each regime, define the regime-wise best feasible stationary policy (informally: the best "target price" consistent with the hard bounds, coupled with the fastest feasible approach under $|p_t - p_{t-1}| \le \Delta$). The dynamic regret is

$$\mathrm{Reg}_T := \sum_{t=1}^{T} \Big(\mathbb{E}[\pi_t^*(r_t)] - \mathbb{E}[\pi_t] \Big), \qquad \pi_t = (p_t - c(x_t))y_t,$$

where $\pi_t^*(r_t)$ denotes the benchmark profit in the current regime and expectations integrate over demand noise and any algorithmic randomness. In contrast to static regret, Reg_T is permitted to depend on the number of regimes R; the goal is to ensure that the dependence is on *change frequency and detectability*, rather than linearly on T.

Decomposing performance losses: learning vs. nonstationarity. The structure of the problem suggests a natural accounting identity. Fix any change-point procedure that triggers a reset at (possibly random) times. Between resets, the algorithm behaves like a stationary learner, so its loss is governed by standard exploration–exploitation tradeoffs. Near change points, however, two additional costs appear: (i) *detection delay*, during which the algorithm keeps using a stale model, and (ii) *false alarms*, which needlessly discard information and restart learning. We summarize both by a detection/reset cost term d_{det}, measured in periods.

Formally, let $d_{\mathrm{det},r}$ denote the number of periods after τ_r until the first reset that "catches" the change to regime r (with $d_{\mathrm{det},1} = 0$ by convention). Let $\bar{\pi}$ be a uniform bound on the per-period profit gap between any two feasible prices, e.g.,

$$\bar{\pi} \ge \sup_{t \le T} \sup_{p,p' \in \mathcal{A}_t(x_t, p_{t-1})} \Big| \mathbb{E}[(p - c(x_t))y \mid p, x_t, r_t] - \mathbb{E}[(p' - c(x_t))y \mid p', x_t, r_t] \Big|.$$

Then the nonstationarity penalty can be controlled by $d_{\mathrm{det},r}\bar{\pi}$, because each period of delay can lose at most $\bar{\pi}$ relative to the regime-wise benchmark.

Proposition (regret decomposition). Suppose that on each regime interval $\mathcal{I}_r$ (excluding a detection window of length $d_{\mathrm{det},r}$ after τ_r), the algorithm attains within-regime regret at most $\tilde{O}(\sqrt{L_r})$ against the best feasible stationary policy for that regime. Then

$$\mathrm{Reg}_T \le \sum_{r=1}^{R} \tilde{O}(\sqrt{L_r}) \;+\; O\Big(\sum_{r=2}^{R} d_{\mathrm{det},r}\,\bar{\pi}\Big).$$

The proof is a partition argument: split time into "well-specified" blocks where the regime is constant and the algorithm has reset, and "boundary" blocks whose total length is $\sum_{r\ge 2} d_{\mathrm{det},r}$. Apply the stationary regret guarantee on the first type and the crude worst-case bound $\bar{\pi}$ on the second type.

What the bound says (and what it does not). The first term, $\sum_r \tilde{O}(\sqrt{L_r})$, is the cost of learning within regimes; it is sublinear in each regime length and therefore scales like $\tilde{O}(\sqrt{T})$ when R is fixed. The second term is the cost of *tracking* changes; it scales approximately linearly in the number of changes, but crucially through the detection delays $d_{\mathrm{det},r}$ rather than through T. This is the sense in which piecewise stationarity is beneficial: if changes are infrequent and quickly detectable, the penalty for nonstationarity is small.

This decomposition also clarifies why stability constraints matter for performance even when they are "free" from a compliance standpoint. If Δ is small, then the benchmark itself may take multiple periods to reach a new target price after τ_r, which reduces $\bar{\pi}$ and can soften the boundary cost. At the same time, a small Δ can slow down the algorithm's ability to explore a range of prices (because the action set effectively contracts around p_{t-1}), which increases within-regime learning constants. Our regret expression does not resolve this governance tradeoff; it simply isolates where Δ enters.

Explicit dependence on noise and drift magnitude. To turn the abstract term $d_{\mathrm{det},r}$ into something interpretable, we need a statistical model of detectability. A canonical special case is linear demand within each regime,

$$D_r(p, x) = a_r(x) - b_r(x)p,$$

with unknown (a_r, b_r) and σ-sub-Gaussian demand noise. In such models, residual-based change-point tests yield detection delays that scale inversely with the squared jump size. Let κ_r denote the minimum magnitude of the

parameter shift at τ_r (in the norm relevant for prediction errors on the realized price/context sequence). Standard concentration arguments then give

$$d_{\mathrm{det},r} = \tilde{O}\Big(\frac{\sigma^2}{\kappa_r^2}\Big),$$

up to logarithmic factors in T and $1/\delta$. Intuitively, larger σ blurs the signal and requires more samples to conclude that the old model no longer fits, while larger κ_r makes the change statistically obvious and shortens the delay.

Combining this detectability scaling with the decomposition yields a regret bound of the form

$$\mathrm{Reg}_T \le \tilde{O}\Big(\sum_{r=1}^{R} \sqrt{L_r}\Big) + \tilde{O}\Big(\sum_{r=2}^{R} \frac{\sigma^2}{\kappa_r^2}\Big),$$

where the second term should be read as "nonstationarity cost is approximately additive across change points, and each change point is cheaper when it is larger and cleaner." The explicit appearance of σ and κ_r is important for practice: it maps directly to engineering choices (smoothing, aggregation, experimentation) and market conditions (demand volatility, abruptness of shocks).

Role of the regime count R. In the worst case, $\sum_{r=1}^{R} \sqrt{L_r} \le \sqrt{RT}$, so the learning term degrades gracefully as regimes become more frequent. The detection term, however, is essentially linear in the number of change points (unless the jumps become easier to detect as they become more frequent). This aligns with economic intuition: each structural break forces a "tax" of re-estimation and re-optimization, even if the algorithm is otherwise efficient.

Limitations and boundary cases. Two boundary cases deserve emphasis. First, if regimes drift gradually so that κ_r is effectively tiny at each τ_r, then change points become statistically indistinguishable from noise and any detector must incur long delays (or many false alarms). In this case, the second term can dominate and the piecewise-stationary abstraction loses predictive power. Second, if the demand model class is misspecified (e.g., the true D_r is nonlinear while we fit linear demand), then within-regime regret should be interpreted as regret relative to the best approximation within the model class; our decomposition still holds, but the benchmark is no longer the true profit-maximizing feasible policy.

Taken together, Theory II formalizes the practical promise of OSP: when the environment is stable most of the time and changes are sufficiently salient relative to noise, we can achieve near-stationary learning rates within regimes and pay only a localized, interpretable penalty around regime boundaries.

8 Practical Evaluation: Off-Policy Evaluation with Guardrails, Counterfactual Bounds, and Stress Tests

A regret bound is only valuable if it can be translated into a deployment protocol that is legible to practitioners: we must be able to (i) evaluate candidate policies before exposing customers to them, (ii) quantify the residual uncertainty in that evaluation, and (iii) demonstrate that compliance constraints are satisfied *mechanically* rather than by good intentions. In our setting, the central practical difficulty is that pricing is *path dependent*: the stability constraint couples decisions across time, so a counterfactual policy cannot be evaluated by period-by-period substitution alone. Our evaluation design therefore treats pricing as a sequential decision rule with state (x_t, p_{t-1}), and it separates what must be guaranteed (hard constraints) from what can only be estimated (revenue and profit impacts).

Logged data requirements and "guardrail-aware" policy definitions. We assume access to historical logs of (x_t, p_t, y_t) under some behavior policy (a human rule, a legacy system, or a randomized experiment). Because our implemented price is always of the form

$$p_t = \Pi_{\mathcal{A}_t(x_t, p_{t-1})}(\tilde{p}_t),$$

any candidate policy can be represented as a *candidate generator* $\tilde{p}_t = \phi_\theta(h_t)$ (with history h_t) plus the same auditable projection layer. This representation is not cosmetic: it ensures that every offline evaluation is conducted on the *actual deployed object* (the projected action), which eliminates a common failure mode where offline performance is reported for an unconstrained policy that will never be implemented. It also makes compliance verifiable by inspection of code and logs.

Off-policy evaluation (OPE) as sequential replay with feasibility. Given a candidate policy π (equivalently ϕ_θ plus projection), the most direct estimate of its revenue is a model-based sequential replay: we take the realized context sequence $(x_1, \ldots, x_T)$ from logs, initialize at the observed p_0, generate the counterfactual price path $\{p_t^\pi\}$ via the policy and the projection operator, and then impute counterfactual demand and profit using an estimated demand model $\widehat{D}(p, x)$ (or regime-indexed $\widehat{D}_r$ when change points are modeled). This produces an estimated counterfactual profit

$$\widehat{\Pi}(\pi) \;=\; \sum_{t=1}^{T} (p_t^\pi - c(x_t))\, \widehat{D}(p_t^\pi, x_t),$$

which is fast to compute and automatically respects the step-size constraint because the path is generated recursively. The limitation is familiar: model-based OPE inherits bias from misspecification, especially when the counterfactual prices fall outside the support of historically observed prices for similar contexts.

When the behavior policy has known randomization (or can be approximated by a propensity model), we can complement the model-based estimate with importance-weighted or doubly robust estimators. The stability constraint simply enlarges the "context" to include the previous price, so each period has conditioning variable $s_t = (x_t, p_{t-1})$ and action $a_t = p_t$. In finite-horizon form, weighted importance sampling uses ratios of action probabilities $\pi(a_t \mid s_t)/\mu(a_t \mid s_t)$, while doubly robust estimators combine a reward model with these ratios to reduce bias. In practice, we recommend using these estimators primarily as *diagnostics*: their variance can be large when the candidate policy deviates materially from historical behavior, which is precisely when one most wants reliable answers. This motivates a conservative reporting style: we treat any OPE point estimate as suggestive, and we emphasize uncertainty quantification and lower bounds.

Counterfactual bounds and "safe improvement" gates. To make offline evaluation decision-relevant, we report not only $\widehat{\Pi}(\pi)$ but also a conservative lower confidence bound. There are two complementary approaches.

First, under parametric demand models (e.g. linear demand in each regime), we can propagate a confidence set for parameters through the counterfactual simulation. Concretely, if θ denotes demand parameters and $\Theta_t(\delta)$ is a high-probability confidence region built from logged data, we compute

$$\underline{\Pi}(\pi) \;=\; \min_{\theta \in \Theta(\delta)} \sum_{t=1}^{T} (p_t^\pi - c(x_t))\, D_\theta(p_t^\pi, x_t),$$

which yields a worst-case profit consistent with the data at level $1-\delta$. This produces an explicit "do-no-harm" gate: we only consider deployment if $\underline{\Pi}(\pi)$ exceeds the estimated profit of the status quo policy by a material margin. The logic is managerial rather than purely statistical: because pricing affects customers and may trigger scrutiny, we require evidence of improvement under plausible demand realizations, not only under the best-fitting model.

Second, when we do not trust parametric structure, we can report partial-identification bounds based on overlap and smoothness assumptions. For example, if we have reliable estimates of demand only on a price interval $\mathcal{P}_{\text{supp}}(x)$ for each context cell, then any counterfactual price $p_t^\pi \notin \mathcal{P}_{\text{supp}}(x_t)$ is flagged as extrapolation. A simple bound is obtained by clipping evaluation to the supported region (a "supported-policy" proxy), while a more informative bound can be obtained under a Lipschitz condition on $D(p, x)$ in

p. Either way, the key output is not a single number but a *range* that makes explicit how much of the purported uplift is coming from extrapolation.

Stress tests with simulated regime shifts. Because nonstationarity is the primary operational risk, we recommend a battery of stress tests that are explicitly adversarial to the learned policy. Starting from a fitted regime-wise demand model, we generate synthetic sequences with controlled change points: shifts in intercepts (demand level), slopes (price sensitivity), and context coefficients (seasonality or traffic effects). We then run the full online algorithm (including the change-point detector and resets) on these synthetic streams, using the same hard-constraint projection, and we record (i) cumulative profit relative to the regime-wise feasible benchmark, (ii) detection delay and false-alarm frequency, and (iii) the distribution of realized price paths (how often the policy sits at caps/floors, how often it is step-limited by Δ). Varying jump sizes, noise levels, and change frequencies produces an empirical "phase diagram" that complements the theory: it shows where the detector reliably localizes breaks, and where the environment is effectively drifting and resets become unstable.

A useful design principle is to include *policy-induced* stress: we simulate not only exogenous parameter jumps but also situations where the policy itself changes the realized price distribution (e.g. it tends to price higher, reducing the data available at low prices). This matters because change-point tests based on residuals can fail when the policy stops visiting informative regions of the action space. In practice, we therefore test the detector under multiple exploration intensities and include "canary" perturbations (small, temporary price probes within the feasible set) to maintain identifiability.

Reporting: compliance is a metric, but violations are structurally zero. We report two families of metrics. The first family is *compliance*: the number of cap/floor violations and step-size violations is identically zero by design, but we still log and report audit-friendly statistics that indicate how binding the constraints are. Examples include the fraction of periods in which projection is active, the average magnitude of clipping, the fraction of periods at $\underline{p}(x_t)$ or $\overline{p}(x_t)$, and the fraction of periods in which the step constraint binds (i.e. $|p_t - p_{t-1}| = \Delta$). These quantities are operationally important because they reveal when the learned policy is "pushing against governance," which can signal either genuine profitability opportunities or model error.

The second family is *revenue and profit*: average profit per period, gross revenue, units sold, and decompositions into margin versus volume effects, all reported with uncertainty intervals. Because profit is the objective but revenue is often the KPI, we report both and explicitly show how results depend on cost $c(x_t)$. For decision-making, we place particular weight on

conservative estimates (lower bounds) and on robustness across stress scenarios, rather than on the single best offline point estimate.

Taken together, these evaluation steps operationalize our central claim: hard constraints can be enforced deterministically, while performance is assessed probabilistically with explicit uncertainty and robustness checks. This creates a deployment pathway in which governance is non-negotiable, learning is incremental, and failures are anticipated through structured stress rather than discovered in production.

9 Extensions: Multi-SKU Coupling, Inventory/Stockouts, CVaR Risk Constraints, and Limited Personalization with Groupwise Constraints

Our baseline model isolates a single product and a one-dimensional action (the posted price). Many revenue-management deployments, however, confront *coupled* decisions: prices interact across products, inventory creates a hard intertemporal state constraint, managers care about downside risk rather than expected profit alone, and personalization is permitted only within explicit groupwise governance rules. We briefly sketch how our framework extends to each of these settings. The common theme is that the *guardrail layer* remains conceptually clean—we still implement actions by projecting a candidate onto an auditable feasible set—but the *inner optimization and learning* typically become multi-dimensional and, in practice, require numerical methods.

Multi-SKU pricing with cross-price effects. Let $k \in \{1, \dots, K\}$ index SKUs and write the action as a price vector $p_t \in \mathbb{R}^K$. A natural generalization of our feasibility constraints is a convex set

$$\mathcal{A}_t(x_t, p_{t-1}) \;=\; \Big\{ p \in \mathbb{R}^K \,:\, \underline{p}_k(x_t) \le p_k \le \overline{p}_k(x_t)\ \forall k,\ \|p - p_{t-1}\|_\infty \le \Delta \Big\},$$

or, when step-size limits are SKU-specific, $\|p - p_{t-1}\|_\infty \le \Delta$ is replaced by $|p_k - p_{t-1,k}| \le \Delta_k$. The implemented action remains

$$p_t \;=\; \Pi_{\mathcal{A}_t(x_t, p_{t-1})}(\tilde{p}_t),$$

now interpreted as Euclidean projection in $\mathbb{R}^K$. When the feasible set is a box with ℓ_∞ step limits, projection decomposes coordinate-wise and remains trivial; when we add *coupling constraints* (e.g. category margin floors, relative-price ladders, MAP policies, or platform rules such as $p_i \le p_j + \Gamma$), the projection becomes a small convex program that can still be solved quickly and logged for audit.

Demand coupling enters through a vector demand function $D_r(p, x) \in \mathbb{R}_+^K$, with profit

$$\pi_t \;=\; \sum_{k=1}^{K} (p_{t,k} - c_k(x_t))\, y_{t,k}, \qquad \mathbb{E}[y_t \mid p_t, x_t, r_t] = D_{r_t}(p_t, x_t).$$

A tractable special case is linear cross-price demand,

$$D_r(p, x) \;=\; a_r(x) - B_r(x)\, p,$$

where $B_r(x)$ is a (typically diagonally dominant) matrix of own- and cross-price sensitivities. Then expected profit is a concave quadratic if $B_r(x)$ is positive semidefinite in the appropriate sense, and computing the myopic constrained optimum reduces to a quadratic program. The learning problem becomes a multivariate regression with change points in (a_r, B_r), and the main practical distinction from the scalar case is not conceptual but computational: we require regularization (to control dimensionality and ensure stability) and numerical solvers for the per-period optimistic or robust optimization step.

Inventory dynamics, stockouts, and censored demand. Inventory introduces a hard state variable that directly couples decisions across time. Let I_t denote on-hand inventory at the start of period t and suppose sales satisfy $y_t \le I_t$ with inventory dynamics

$$I_{t+1} \;=\; I_t - y_t + u_t,$$

where u_t is an exogenous replenishment (possibly zero for a finite-season problem). Two complications arise. First, the feasible action set may now depend on inventory via governance rules (e.g. "do not discount below $p^{\min}(x)$ when I_t is scarce"), creating $\mathcal{A}_t(x_t, p_{t-1}, I_t)$. Second, the retailer may observe only *censored* demand: when I_t is low, observed sales equal $\min\{I_t, \text{latent demand}\}$, so residual-based change-point detection and demand estimation must explicitly account for censoring.

Conceptually, one can treat (x_t, p_{t-1}, I_t) as the observed state and interpret pricing as a constrained control problem with unknown transition law (because the demand model is unknown). The same "candidate plus projection" architecture applies mechanically, but optimality is no longer myopic: prices trade off current margin against preserving inventory for higher-value future periods. In regimes with stable demand, approximate dynamic programming or model predictive control can be used: we (i) fit a demand model, (ii) solve a finite-horizon stochastic program with inventory dynamics to generate a candidate $\tilde{p}_t$, and (iii) project to enforce hard constraints. In piecewise-stationary environments, the detector-and-reset logic continues to be useful, but detection statistics should be built from *uncensored* periods

(when I_t is comfortably above realized sales) or from likelihoods that incorporate truncation. In practice, these extensions are feasible but are rarely closed-form; numerical planning is the norm.

Risk-sensitive objectives and CVaR constraints. Managers often care about downside outcomes (e.g. weekly profit shortfalls) due to budget targets, reputational concerns, or regulatory scrutiny. A convenient formalization is to impose a conditional value-at-risk constraint on a loss functional. For instance, for an evaluation window of length H, define cumulative profit $S = \sum_{t=1}^{H} \pi_t$ and loss $L = -S$. A CVaR constraint takes the form

$$\mathrm{CVaR}_\alpha(L) \;\le\; \rho,$$

for confidence level $\alpha \in (0,1)$ and tolerance ρ. Using the standard variational representation,

$$\mathrm{CVaR}_\alpha(L) \;=\; \min_{\eta \in \mathbb{R}} \left\{ \eta + \frac{1}{1-\alpha}\, \mathbb{E}\big[(L-\eta)_+\big] \right\},$$

we can incorporate risk either as (i) a constraint enforced via a Lagrangian primal–dual update or (ii) a penalty added to expected profit. The guardrail logic remains unchanged for *price* constraints, but we now add a *statistical* guardrail on tail outcomes: offline, we estimate a conservative upper confidence bound for $\mathrm{CVaR}_\alpha(L)$ under the candidate policy using sequential replay with demand uncertainty; online, we can maintain a running estimate of tail losses and tighten exploration when the estimated risk budget is being consumed.

The key limitation is that CVaR is inherently distributional and thus less amenable to high-probability guarantees under arbitrary nonstationarity. In particular, change points can concentrate losses exactly when the policy is least calibrated. Practically, this makes stress testing (with adversarial regime shifts) even more central, and it shifts attention from asymptotic regret constants toward finite-sample tail diagnostics.

Limited personalization with groupwise constraints. Finally, consider limited personalization in which customers are partitioned into observable groups $g \in \{1, \dots, G\}$ (e.g. geography, loyalty tier, platform channel), with potentially different demand responses $D_{r,g}(p, x)$. The retailer may post group-specific prices $p_{t,g}$, but governance often imposes explicit restrictions such as bounded dispersion,

$$|p_{t,g} - p_{t,g'}| \le \Gamma \quad \forall g, g',$$

or monotonicity rules (e.g. premium tiers cannot face lower prices than standard tiers), in addition to caps/floors and step-size limits for each group.

These are naturally encoded as a convex feasible set in $\mathbb{R}^G$, so the implemented pricing vector again takes the projected form

$$p_t \;=\; \Pi_{\mathcal{A}_t(x_t,p_{t-1})}(\tilde{p}_t),$$

where $\mathcal{A}_t$ includes both per-group and cross-group constraints. This representation is operationally attractive: it ensures that any personalization logic, no matter how complex, cannot violate explicit non-discrimination guardrails.

From a learning perspective, the main new issue is data sparsity: each group provides fewer observations, yet the algorithm must detect regime shifts that may be group-specific. A pragmatic compromise is to adopt a hierarchical structure (shared components across groups plus group deviations) and to run change-point detection at multiple resolutions: global tests to detect market-wide shifts and group-level tests to detect localized breaks. The corresponding optimization step (optimistic, robust, or risk-constrained) typically becomes a small but nontrivial convex program each period, again favoring numerical methods with warm starts and strict logging for audit.

Why numerical methods are the rule, not the exception. Across these extensions, the economics is unchanged: we still face a tradeoff between adapting to demand and honoring operational constraints that are non-negotiable. What changes is that the inner problems cease to be one-dimensional and closed-form. In our view, this is not a weakness of the framework but a realistic boundary: in deployed systems, transparency and enforceability come from the *projection-based guardrails*, while performance comes from whatever estimation and numerical optimization is credible under the data and governance environment. Our model clarifies where one can demand certainty (constraint satisfaction) and where one must instead manage uncertainty (profit, risk, and nonstationarity) through conservative bounds and stress-tested numerical procedures.

10 Conclusion: Implications for Revenue Management, Auditing, and Policy; Limitations and Next Steps

We have studied dynamic pricing in the environment that practitioners repeatedly describe as "the hard part": demand is uncertain and can shift abruptly, while the permissible prices are governed by *non-negotiable* operational and policy constraints. Our main message is that these two features can be separated cleanly. Constraint satisfaction is an *engineering* requirement that should be guaranteed mechanically (via projection onto an explicit feasible set), whereas adaptation to demand is a *statistical* problem that should be managed with learning and detection tools that come

with interpretable performance guarantees. The resulting architecture is conceptually simple: propose a candidate price using any reasonable forecasting/optimization logic, then pass it through an auditable "guardrail layer" that enforces caps, floors, and step-size limits by construction.

For revenue management, the key implication is that stability constraints and price bounds need not be treated as afterthoughts that invalidate theory; rather, they can be incorporated as first-class primitives. The feasible set $\mathcal{A}_t(x_t, p_{t-1})$ is not merely a modeling convenience: it matches how pricing teams actually operate, with hard limits imposed by brand policy, platform rules, and compliance. Once we accept these guardrails as binding, the relevant performance benchmark changes as well. The natural comparator is not the unconstrained static monopoly price, but the *best feasible stationary policy within each regime*—including the fact that even an omniscient benchmark must obey the same caps, floors, and step-size restrictions. This shift in benchmark matters operationally: it aligns evaluation with what a compliance team will accept, and it makes regret (and its decomposition) a credible diagnostic for how quickly a system adapts following a demand break.

A second implication is interpretability of "why profits fell" in the presence of nonstationarity. When demand is piecewise stationary, our dynamic regret bound decomposes into within-regime learning terms (scaling like $\tilde{O}(\sqrt{L_r})$) plus boundary losses induced by detection and reset. In plain terms, performance degradation has two sources: *estimation error* when the regime is stable, and *calibration lag* when the regime changes. This decomposition suggests practical KPIs that mirror the theory: (i) within-regime fit and exploration diagnostics (residual variance, parameter confidence widths, effective sample sizes) and (ii) change-point KPIs (average detection delay, false-alarm rate, and the profit impact in the post-change window). Importantly, stability constraints interact with these KPIs in a predictable way: they can modestly reduce short-run responsiveness after a break, yet they also reduce harmful variance and limit extreme actions when the model is misspecified.

The auditing implications follow directly from the projection-based design. Because the implemented price satisfies $p_t \in [\underline{p}(x_t), \overline{p}(x_t)]$ and $|p_t - p_{t-1}| \leq \Delta$ for all t by construction, compliance can be verified ex post without debating statistical assumptions about demand. Moreover, the projection step is itself loggable: for each period one can record the candidate $\tilde{p}_t$, the feasible interval $\mathcal{A}_t(x_t, p_{t-1})$, and the final implemented $p_t = \Pi_{\mathcal{A}_t}(\tilde{p}_t)$. This creates a transparent trail that distinguishes "the model wanted to do X" from "the guardrail allowed only Y," which is precisely what internal risk committees and external regulators often demand. In our view, this distinction is also economically meaningful: it clarifies that the opportunity cost of governance is not a vague loss of "flexibility," but the measurable gap between unconstrained and constrained optima, plus the transitional loss imposed by

step-size limits.

From a policy perspective, the framework provides a disciplined way to think about rules such as anti-gouging statutes, platform price caps, and fairness constraints. Such rules are frequently criticized for "distorting prices," yet in deployed systems they are unavoidable constraints that can be implemented either opaquely (through ad hoc overrides) or transparently (through explicit feasible sets). Our approach advocates the latter: encode policy in $\underline{p}(\cdot)$, $\overline{p}(\cdot)$, and Δ (and, in multi-dimensional settings, in convex coupling constraints), then quantify the induced welfare or profit loss relative to the *policy-feasible* benchmark. This reframing helps avoid a common category error: comparing outcomes under a governed system to an infeasible counterfactual. It also highlights a constructive design margin for regulators and platforms: stability bounds and caps can be tuned to achieve compliance goals while keeping the induced loss small when the objective is locally flat near the optimum, consistent with second-order "flat-top" logic.

These benefits come with limitations that delimit what our guarantees do *not* say. First, piecewise stationarity is an approximation. Real markets may drift gradually, exhibit seasonality not fully captured by x_t, or respond endogenously to the firm's own history. In such environments, change-point detection may be triggered by unmodeled cyclicality, and resets may discard useful information. Second, we have assumed that marginal cost $c(x)$ and the guardrails are known and correctly specified; in practice, costs can be measured with error and policy constraints can change discretely, which can look statistically like demand breaks. Third, our regret bounds are derived under specific noise and model classes (e.g. sub-Gaussian shocks, linear demand in the tractable case). Heavy tails, strategic stockpiling, and unobserved rationing can all violate these assumptions and degrade both inference and detection. Finally, we have abstracted from competition and strategic consumers; in many categories, demand depends on rivals' prices and on customers' expectations, and those feedback effects can invalidate a purely exogenous demand-regime interpretation.

These limitations point to several next steps. On the modeling side, an important extension is to integrate competition: demand regimes may be induced by rivals' policy changes, entry/exit, or algorithmic responses, and the "breaks" may be partially predictable from public signals. On the statistical side, it is natural to replace sharp resets with *adaptive forgetting* or model averaging, and to develop robust detection methods that distinguish parameter shifts from volatility bursts or censoring. On the governance side, we see a need for formal "auditability metrics" that sit alongside regret: for example, measures of how often the projection binds, how sensitive outcomes are to the choice of Δ, and how frequently groupwise constraints (in limited personalization) become active. Finally, we expect that practice will increasingly demand *stress-tested* guarantees—not only average-case regret under a stochastic model, but also performance under adversarial or worst-case

sequences that encode crisis periods.

Stepping back, the central tradeoff our model illuminates is not merely between exploration and exploitation, but between *adaptation and enforceability*. Revenue-management systems succeed when they can learn quickly *and* remain legible to the institutions that must sign off on their actions. Projection-based guardrails offer a principled way to draw that boundary: we can demand certainty where it is operationally required (zero constraint violations), and we can manage uncertainty where it is inevitable (demand estimation and nonstationarity) using tools that yield transparent diagnostics and interpretable regret decompositions. In this sense, the framework is less a prescription for one specific algorithm than a blueprint for building pricing systems that are simultaneously performant, governable, and auditable.