Tracking Demand Drift in Dynamic Pricing via Sliding-Window Gaussian-Process UCB

Liz Lemma Future Detective

January 16, 2026

Abstract

Dynamic pricing in modern digital markets is inherently nonstationary: competitor repricing bots, shifting consumer sentiment, and platform design changes can move the demand curve on the time scale of days or hours. The source paper (Ananda–Agrawala–Bodas, 2025) shows that GP-UCB/Bayesian Optimization (BO-Inf) yields strong regret guarantees in stationary, infinite-inventory pricing, and proposes GP-based methods for finite inventory. This paper pushes that line to the 2026 reality by studying nonstationary expected revenue functions. We model expected revenue at time t as an unknown function f_t(p) in an RKHS that drifts over time with a bounded variation budget. We propose a sliding-window (or exponentially-discounted) GP-UCB pricing rule that uses only recent data to maintain calibrated uncertainty, enabling tracking of a moving optimum. We provide high-probability dynamic regret bounds that decompose cleanly into (i) a learning term governed by GP information gain over the effective window and (ii) a drift term governed by the variation budget. We derive an optimal window choice and show how performance depends on drift magnitude, kernel smoothness, and observational noise. Extensive experiments with seasonal drift, abrupt regime shifts, and benchmark pricing environments demonstrate large gains over stationary BO-Inf and deep RL baselines in low-data regimes, while requiring fewer price changes and no parametric demand assumptions.

Table of Contents

3. 3. Model: single-product continuous pricing; time-varying expected revenue f_t; noise model; drift/variation budget definitions; benchmarks (dynamic oracle vs constrained oracle).

4. 4. Algorithm: Sliding-Window GP-UCB (SW-GP-UCB) and Exponentially-Discounted GP-UCB (ED-GP-UCB); implementation details (continuous argmax, discretization, hyperparameter handling).

5. 5. Theory I (confidence): concentration / confidence sets for windowed GP posterior under drift; explicit separation into estimation uncertainty + drift bias term.

6. 6. Theory II (dynamic regret): main theorem bounding dynamic regret; corollaries for squared-exponential kernel in 1D; window optimization; discussion of rates vs V_T.

7. 7. Extensions: (i) switching-cost / limited price changes via batching; (ii) seasonal drift structure (periodic V_T); (iii) contextual drift (adding covariates x_t).

8. 8. Experiments: synthetic drift (sinusoidal, piecewise constant, adversarial bounded-variation), repricing-bot stylized setting; comparisons to stationary BO-Inf, PPO/SAC, and simple heuristics; sensitivity to W and discount factor.

9. 9. Discussion & limitations: kernel misspecification, hyperparameter drift, partial observability; when the drift model is violated; policy/operational implications.

10. 10. Conclusion: practical takeaways for nonstationary pricing; open problems (multi-agent competition, fairness constraints, finite-inventory integration).

1 Introduction

Digital pricing in 2026 operates in an environment where the classical assumption of stationarity—that the mapping from price to expected revenue is stable over time—is increasingly hard to defend even as a useful approximation. On large retail platforms, prices are updated by automated repricing agents that react within minutes to competitors, inventory signals, and ranking incentives. Platforms themselves frequently adjust fee schedules, delivery promises, and search or recommendation rules, all of which shift conversion rates at a fixed posted price. Meanwhile, marketing channels and consumer attention are mediated by auctions and algorithms whose parameters are updated continuously, creating demand conditions that can drift even when the underlying product is unchanged. In such settings, a pricing policy that treats the demand curve as fixed is not merely misspecified in a statistical sense; it can be systematically slow to respond to changes that are economically meaningful and strategically induced.

This motivates a modeling stance in which the seller faces a sequence of expected revenue functions that evolve over time. The key empirical feature is not that changes are arbitrary, but that they are neither perfectly predictable nor negligible. Some shifts are gradual (e.g., a steady deterioration in conversion as a product becomes less novel), others are episodic (e.g., a platform UI change or a competitor entering), and many are structured (weekday/weekend patterns, seasonal promotions). A seller who learns a single "best" price over a long horizon risks exploiting a price that was optimal for yesterday's environment. Conversely, a seller who reacts too aggressively to short-run noise risks chasing transient fluctuations and leaving revenue on the table. Our goal is to formalize this economic tradeoff and provide an algorithmic prescription that is both implementable and backed by performance guarantees that explicitly account for nonstationarity.

A convenient starting point in modern dynamic pricing is the Bayesian optimization view: treat the expected revenue as an unknown function of price, place a Gaussian process prior over this function, and choose prices using an acquisition rule that trades off exploration and exploitation. In the infinite-inventory baseline, the seller observes noisy realizations of revenue (or an unbiased proxy) after posting each price. Under stationarity, GP-UCB style policies deliver regret guarantees that scale essentially as $\tilde{O}(\sqrt{T})$ up to information-gain factors, and recent pricing papers (including infinite-inventory Bayesian optimization formulations, sometimes referred to as BO-Inf) leverage exactly this logic to justify sequential experimentation with minimal parametric structure. However, these guarantees are fundamentally anchored to a fixed objective: they compare performance to the single best price in hindsight for one underlying function. When the revenue curve itself moves, a policy can have small *static* regret while still persistently lagging the moving optimum. The performance yardstick must therefore change with

the environment.

We adopt *dynamic regret* as the appropriate benchmark for nonstationary pricing: we compare the seller's sequence of posted prices to the sequence of period-by-period optimal prices that a clairvoyant would choose if they knew the current revenue function. This benchmark captures the operational objective faced by practitioners: not to find a timeless "optimal price," but to track the best current price as market conditions evolve. To make this meaningful, we require a discipline on how fast the environment can move. Economically, it is implausible that the entire revenue curve can change arbitrarily each period without bound; even in fast-moving platforms, frictions in consumer attention, shipping times, and competitive adjustment limit how violently demand can shift. We encode this idea through a variation budget that bounds the cumulative drift of the revenue function over the horizon. This is deliberately weak: it neither imposes a parametric law of motion nor requires the seller to observe the drift, but it rules out environments that are so adversarial that no learning-and-tracking policy could perform well.

On the statistical side, we preserve the nonparametric flexibility that makes GP methods attractive in pricing applications. Rather than specifying a particular demand model (e.g., logit with time-varying coefficients), we assume only that each period's expected revenue function is smooth in price in the sense of belonging to a reproducing kernel Hilbert space associated with a squared-exponential kernel, with a uniform norm bound. This assumption is a formal way to state that small price changes do not cause arbitrarily large expected revenue changes, a property that is often consistent with observed conversion curves after appropriate normalization and price scaling. Observations are allowed to be noisy and heteroskedastic at the level of realized demand and revenue; the analysis uses a standard sub-Gaussian noise condition, which can be interpreted as ruling out extremely heavy-tailed shocks after basic winsorization or aggregation (common in platform analytics). Together, these primitives yield a model that is economically interpretable, empirically plausible, and mathematically tractable.

Our main contribution is to show that a simple modification of stationary GP-UCB—restricting learning to *recent* data—achieves a principled tracking guarantee in this nonstationary pricing environment. The algorithm we study forms a Gaussian process posterior using only a sliding window of the last W price–feedback observations and then chooses the next price by maximizing an upper confidence index. The window length W becomes the key design parameter: a longer window reduces statistical uncertainty by pooling more data, but increases the risk of using stale observations that reflect outdated demand conditions. A shorter window adapts more quickly to drift but is noisier and can lead to under-exploration of profitable price regions. This "memory" choice is exactly the operational question faced by pricing teams when they decide how far back to look in training data for forecasting or experimentation.

We provide a dynamic regret bound that makes this bias–variance trade-off explicit and yields a transparent tuning rule. In our bound, one term decreases with W and captures statistical learning difficulty (through the kernel information gain), while the other increases with W and captures drift-induced mismatch. Balancing these terms delivers a window choice that adapts to the magnitude of nonstationarity: more stable environments justify longer memory and near-stationary performance, while rapidly changing environments call for shorter memory and faster tracking. Beyond the theoretical value, this structure offers a practical guideline: by estimating drift proxies (e.g., rolling residual instability or the frequency of platform changes), one can select a window that is aligned with the prevailing market regime rather than fixed ex ante.

Finally, we emphasize limitations and scope. Our framework does not claim that platform-induced changes are exogenous; in many markets, competitor bots and platform policies respond strategically to prices, and the resulting dynamics may violate the bounded-variation condition in extreme episodes. Likewise, the infinite-inventory baseline abstracts from stockouts and intertemporal substitution, both central in retail. We view these as important extensions, but also as reasons to begin with a transparent revenue-tracking model: it isolates the core learning problem created by nonstationarity and clarifies what guarantees are possible without strong structural assumptions. The upshot is a pricing-oriented nonstationary GP bandit framework that retains the flexibility of Bayesian optimization while replacing stationary regret guarantees with tracking guarantees that are better aligned with how algorithmic pricing is actually used.

2 Related work

Our setting sits at the intersection of three literatures: dynamic pricing under stationarity (typically with parametric demand), Bayesian optimization views of pricing that use Gaussian processes as flexible surrogates, and nonstationary bandits that benchmark performance against a moving target. A unifying theme is the same economic tension that motivates our analysis: a seller wants to exploit what has been learned about the revenue curve while retaining enough adaptivity to remain close to the contemporaneous optimum when the environment moves.

The classical dynamic pricing literature in operations and revenue management largely starts from a stationary demand system with unknown parameters and studies optimal or approximately optimal experimentation policies. Canonical models assume demand arrives as a stochastic process whose intensity depends on price through a parametric function, and the seller updates beliefs over a low-dimensional parameter vector (e.g., linear, logit, or isoelastic demand). This line includes Bayesian formulations and frequentist

learning frameworks, and delivers regret bounds or asymptotic optimality results that reflect how quickly the parameters can be estimated (see, among many others, **????**). The stationarity assumption is not merely technical: it pins down a single "true" demand curve so that long-run exploration pays off. In practice, however, pricing teams frequently retrain demand models on rolling data precisely because a single global fit can become stale; this operational reality is the starting point for our nonstationary benchmark.

A complementary stationary strand replaces parametric structure with bandit-style learning directly over prices. In finite action sets, multi-armed bandit algorithms provide regret guarantees without specifying a demand model, and in continuous action spaces one can exploit Lipschitz or smoothness assumptions to obtain rates that depend on the dimension of the price vector. In pricing applications, these approaches are attractive because they map cleanly to sequential A/B testing over price points, but they typically measure performance relative to a single best price in hindsight. This static benchmark can be economically misleading when the revenue curve shifts: a policy can be "no-regret" in the static sense while persistently charging yesterday's price. Our use of dynamic regret aligns the benchmark with the managerial objective of tracking, rather than learning once and exploiting forever.

Recent work has popularized a Bayesian optimization (BO) perspective for pricing, particularly in online retail settings where the price–revenue relationship is noisy, potentially nonlinear, and costly to model structurally. The idea is to treat expected revenue as a black-box function of price, impose smoothness via a Gaussian process prior, and select prices through an acquisition rule such as upper confidence bounds or Thompson sampling (e.g., GP-UCB and its variants; see **?**). This "BO-Inf" viewpoint is appealing because it produces implementable algorithms with uncertainty quantification and can accommodate heterogeneity through contextual extensions. Yet the standard theoretical guarantees again rest on stationarity: the GP posterior aggregates all past data as if it were generated from one fixed function. When the objective drifts, the posterior can become overconfident in outdated regions, an effect that practitioners often observe as "model inertia" after platform changes or competitor entry.

Nonstationary bandits address exactly this issue by weakening stationarity and strengthening the benchmark. A prominent approach models the environment through a variation budget that bounds cumulative drift of mean rewards, yielding dynamic regret rates of order $T^{2/3}V_T^{1/3}$ (up to logarithmic factors) under suitable conditions; see, for example, **?** and subsequent refinements. Algorithmically, these results motivate forgetting mechanisms such as sliding windows, periodic restarts, or exponential discounting, which mirror common engineering practices in online learning systems. Related models consider piecewise-stationary environments with a bounded number

of changepoints, where one can combine change detection with exploitation within segments. While this literature provides sharp insight into the bias–variance tradeoff created by drift, many results are developed for finite action sets or for structured classes (e.g., linear bandits), and translating them to continuous pricing with nonparametric smoothness requires additional work.

Nonstationary Gaussian process bandits take a step in this direction by allowing the latent function to evolve over time. Existing approaches include (i) treating time as an additional input dimension and placing a separable spatiotemporal kernel over (p, t), (ii) explicitly modeling temporal evolution through state-space or Markovian dynamics over function values, and (iii) adopting algorithmic forgetting (windowing or discounting) while retaining the GP regression machinery. Representative contributions analyze time-varying GP-UCB style policies and derive regret bounds that depend on information-gain quantities associated with the chosen kernel and the effective memory of the algorithm (e.g., **?**). These papers clarify that one can, in principle, track a drifting optimum in nonparametric settings, but they often leave open questions that are central in pricing. First, when time is appended as a covariate, the resulting kernel complexity can obscure tuning: the regret depends on information gain in a higher-dimensional space, and the implied dependence on the time length-scale is hard to map to actionable guidance for how far back one should trust data. Second, several analyses focus on generic function maximization rather than the institutional details of revenue data (e.g., the fact that practitioners often observe realized revenue or conversion, not a direct noisy oracle of expected revenue). Third, while discounting and windowing are widely used heuristics, pricing applications benefit from bounds that isolate the precise economic cost of staleness, because that cost is what determines how aggressively a pricing system should "forget" after a market shift.

Our contribution is best viewed as importing the nonstationary bandit logic—dynamic regret under a variation budget—into the BO-style, continuous-price framework that practitioners increasingly use, and doing so in a way that makes tuning transparent in one-dimensional pricing. By working with a sliding-window GP posterior, we preserve the computational and modeling conveniences of GP regression while directly controlling the mismatch between old observations and the current revenue curve. The resulting bound decomposes cleanly into a statistical term (driven by uncertainty and the information gain over the last W points) and a drift term (linear in $V_T W$), making explicit the operational tradeoff between learning precision and adaptivity. This decomposition also clarifies when sophisticated spatiotemporal kernels are likely to be worthwhile: if drift is largely seasonal or structured, encoding that structure can reduce the effective variation budget, whereas in environments dominated by irregular platform shocks, simple forgetting may be more robust.

Finally, our approach has limitations relative to some strands of the pric-

ing literature. We do not model strategic interaction with competitors or platforms; instead, we treat nonstationarity as an exogenous drift bounded in total variation. This abstraction is deliberate: it yields a tractable performance benchmark and a policy prescription that can serve as a baseline even when richer dynamics are present. Incorporating endogenous drift, inventory constraints, or intertemporal demand substitution would require additional state variables and typically changes the objective from one-period revenue maximization to a dynamic program. We view the present framework as a useful intermediate step: it captures a first-order feature of modern algorithmic pricing—that the revenue curve moves—and provides guarantees that directly speak to the practical question of how much history a pricing algorithm should use when the world does not stand still.

3 Model

We study a single seller who posts a scalar price each period and learns the revenue curve from noisy feedback while the environment drifts over time. Time is indexed by $t = 1, \ldots, T$, and the seller chooses a price p_t from a feasible interval $[p_\ell, p_h] \subset \mathbb{R}_+$. The one-dimensional action space is deliberate: many pricing teams experiment over a single "list price" or a dominant control knob (e.g., a uniform markup), and the key economic tension we want to isolate is intertemporal rather than high-dimensional.

Demand at time t is stochastic and depends on the posted price. Let $D_t(p)$ denote the (random) quantity demanded at time t if the seller were to post price p. In the baseline infinite-inventory setting, realized sales equal realized demand, so $q_t = D_t(p_t)$. Period-t revenue is

$$r_t := p_t q_t = p_t D_t(p_t).$$

We work primarily with the expected revenue function

$$f_t(p) := \mathbb{E}[r_t \mid p_t = p] = \mathbb{E}[p\, D_t(p)]\,,$$

which maps prices to expected revenues and is allowed to vary with t. This formulation absorbs a wide range of underlying demand primitives: shifts in demand levels, changes in price sensitivity, and composition effects in the consumer population all manifest as movement in f_t. From a managerial perspective, f_t is the object a pricing system implicitly estimates when it retrains on transaction data and then optimizes predicted revenue.

The seller does not observe f_t directly. Instead, after choosing p_t, the seller observes a noisy feedback signal y_t that is informative about $f_t(p_t)$. Our baseline measurement model is

$$y_t = f_t(p_t) + \varepsilon_t,$$

where ε_t is conditionally σ-sub-Gaussian given the past (so $\mathbb{E}[\varepsilon_t \mid \mathcal{F}_{t-1}] = 0$ and its tails are controlled uniformly). This abstraction captures two common data regimes. First, if we set $y_t = r_t$, then the noise term reflects demand randomness, unmodeled covariates (promotions, traffic shocks), and any misspecification in mapping transactions to a single scalar outcome. Second, if practitioners use an intermediate metric such as conversion or contribution margin, then y_t may already be a smoothed or debiased proxy; the sub-Gaussian condition remains a convenient way to state that extreme outliers are not too frequent. We emphasize that we do not require a parametric demand model: the only structure is imposed directly on the unknown function $f_t(\cdot)$.

To formalize smoothness in a way that is compatible with Gaussian process regression, we assume each period-t expected revenue function lies in the reproducing kernel Hilbert space (RKHS) associated with a kernel $k(\cdot, \cdot)$. Throughout, we take k to be squared-exponential on $[p_\ell, p_h]$, and we impose a uniform RKHS norm bound:

$$\text{(H1)} \qquad f_t \in \mathcal{H}_k \ \text{ and } \ \|f_t\|_{\mathcal{H}_k} \le B \quad \forall t.$$

Economically, (H1) says that the revenue curve is smooth in price and does not exhibit arbitrarily sharp spikes. In many retail and service settings, this is a reasonable reduced-form approximation: small price changes typically do not create discontinuous jumps in expected revenue absent stockouts or rationing. Methodologically, (H1) is what converts function learning into a tractable nonparametric estimation problem with finite-sample uncertainty quantification.

The central departure from stationary Bayesian optimization is that we allow f_t to drift. Rather than specifying a particular stochastic law of motion for the function, we follow the nonstationary bandit tradition and restrict attention to an environment class characterized by a *variation budget*:

$$\text{(H3)} \qquad V_T := \sum_{t=2}^{T} \sup_{p \in [p_\ell, p_h]} \left| f_t(p) - f_{t-1}(p) \right| \le \bar{V}.$$

The supremum over prices makes the drift notion economically strong: it bounds how much the entire revenue curve can change from one period to the next, not merely the value at the posted price. This strength is useful for robust performance guarantees because the seller must contemplate counterfactual prices when deciding how to explore. At the same time, the budget interpretation remains operational: V_T is small in stable markets where only slow trends occur, and it is large in environments with frequent shocks (competitor entry, platform redesigns, changes in ad auctions). Importantly, V_T is not observed by the seller; it summarizes the difficulty of the instance ex post, and it will govern the attainable tracking rate.

Performance is benchmarked against an oracle that is allowed to change its price as the environment changes. Let

$$p_t^* \in \arg\max_{p \in [p_\ell, p_h]} f_t(p)$$

denote a period-by-period *dynamic oracle* price. Our primary criterion is *dynamic regret*,

$$\mathrm{Reg}_T^{\mathrm{dyn}} := \sum_{t=1}^{T} \big(f_t(p_t^*) - f_t(p_t)\big),$$

which measures the cumulative revenue loss from failing to track the contemporaneous optimum. This benchmark aligns with a common managerial objective: a pricing system should not only learn demand, but also remain close to "the right price today" when the underlying response curve shifts. In contrast, a static benchmark (a single best price in hindsight) can mask economically costly inertia, because it can declare success even when the policy is consistently late in reacting to drift.

In some applications, however, the seller cannot freely jump to the period-by-period maximizer. Prices may be constrained by menu costs, fairness concerns, platform guardrails, or internal governance that limits the frequency or magnitude of changes. To capture this, we also consider *constrained* oracles. One natural variant imposes a bound on price movement, $|p_t - p_{t-1}| \le \eta$, reflecting operational limits on how quickly a pricing team can adjust. Another variant introduces an explicit adjustment cost, either a fixed cost $c\,\mathbf{1}\{p_t \neq p_{t-1}\}$ or a proportional cost $c|p_t - p_{t-1}|$, so the relevant benchmark maximizes $\sum_{t=1}^{T} f_t(p_t)$ net of adjustment costs. These constrained benchmarks are economically appealing when frequent changes themselves are costly, and they clarify a limitation of pure tracking metrics: a policy that perfectly follows p_t^* may be infeasible or undesirable in practice. We keep the dynamic oracle as our baseline because it yields a clean decomposition between statistical uncertainty and staleness from drift, but the constrained oracle interpretation will matter when translating theoretical guidance into deployment rules (e.g., tuning "forgetting" while also imposing change limits).

This model distills the core tradeoff we care about. Because feedback is noisy, the seller would like to pool many observations to reduce uncertainty about $f_t(\cdot)$. Because f_t drifts, old observations can become misleading, so the seller must discount or forget history to avoid overconfidence in an outdated revenue curve. The next section makes this tension algorithmic by specifying GP-based policies that balance exploration, exploitation, and adaptivity through either sliding windows or exponential discounting.

4 Algorithm: windowing and discounting for non-stationary GP-UCB

Our goal is to turn the tradeoff described above—pooling data to reduce noise versus forgetting data to avoid staleness—into a concrete pricing rule. We do so by combining Gaussian process (GP) regression with an "optimism" principle: at each date we choose the price that maximizes an upper confidence bound (UCB) on the contemporaneous revenue curve. The only nonstandard ingredient relative to stationary GP-UCB is that we modify the GP posterior so that older observations receive less weight, either by truncation (a sliding window) or by exponential discounting.

SW-GP-UCB (sliding window). Fix a window length $W \in \{1, \dots, T\}$. At the start of period t, we retain only the most recent data

$$\mathcal{D}_{t-1}^{(W)} := \{(p_s, y_s) : s = \max\{1, t-W\}, \dots, t-1\}.$$

Using $\mathcal{D}_{t-1}^{(W)}$, we compute the standard GP posterior mean and standard deviation, denoted $\mu_{t-1}^{(W)}(\cdot)$ and $\sigma_{t-1}^{(W)}(\cdot)$. Concretely, if we write $\mathbf{p} = (p_{t'})_{t' \in \mathcal{I}_{t-1}}$ and $\mathbf{y} = (y_{t'})_{t' \in \mathcal{I}_{t-1}}$ for the window index set $\mathcal{I}_{t-1} = \{\max\{1, t-W\}, \dots, t-1\}$, and define the kernel matrix $K \in \mathbb{R}^{n \times n}$ with $K_{ij} = k(p_i, p_j)$ (where $n = |\mathcal{I}_{t-1}| \le W$), then with noise variance parameter λ we have for any candidate price p:

$$\mu_{t-1}^{(W)}(p) = \mathbf{k}(p)^\top (K + \lambda I)^{-1} \mathbf{y}, \qquad \left(\sigma_{t-1}^{(W)}(p)\right)^2 = k(p,p) - \mathbf{k}(p)^\top (K + \lambda I)^{-1} \mathbf{k}(p),$$

where $\mathbf{k}(p) = (k(p_i, p))_{i=1}^n$. In our theoretical development, λ plays the role of σ^2 from the sub-Gaussian noise assumption, but in implementations it is best viewed as a ridge parameter that stabilizes inference when observations are nearly collinear in price.

Given $(\mu_{t-1}^{(W)}, \sigma_{t-1}^{(W)})$, SW-GP-UCB selects

$$p_t \in \arg\max_{p \in [p_\ell, p_h]} \; \mu_{t-1}^{(W)}(p) + \kappa_t \, \sigma_{t-1}^{(W)}(p),$$

where κ_t is an exploration multiplier. Intuitively, the mean term exploits what we have learned from recent data, while the standard deviation term forces occasional experimentation in regions where recent data are sparse. After posting p_t, we observe y_t and update the window by adding (p_t, y_t) and dropping (p_{t-W}, y_{t-W}) when applicable.

ED-GP-UCB (exponential discounting). Sliding windows forget abruptly, which can be undesirable when drift is gradual and we would rather down-weight than discard data. Exponential discounting implements a smooth "memory" with a parameter $\rho \in (0, 1)$. At time t, we assign weight

$$w_{t,s} := \rho^{t-s} \qquad \text{to observation } (p_s, y_s), \quad s < t,$$

so that older samples receive geometrically smaller influence. A convenient way to implement this within GP regression is to interpret downweighting as inflating the effective noise variance of older points. Specifically, letting Σ_t be the diagonal matrix with entries $(\Sigma_t)_{ss} = \lambda / w_{t,s}$ for the included indices $s < t$, the discounted posterior takes the same algebraic form as above with $(K + \lambda I)$ replaced by $(K + \Sigma_t)$. The resulting decision rule is again UCB:

$$p_t \in \arg \max_{p \in [p_\ell, p_h]} \mu_{t-1}^{(\rho)}(p) + \kappa_t \, \sigma_{t-1}^{(\rho)}(p),$$

where $\mu_{t-1}^{(\rho)}$ and $\sigma_{t-1}^{(\rho)}$ denote the discounted posterior objects. From a managerial standpoint, ρ plays the role of an exponential moving-average parameter: ρ close to one corresponds to long memory (stable markets), while smaller ρ corresponds to rapid forgetting (high-churn environments). In our later bounds, this mapping is formalized through an "effective window" size $W_{\text{eff}} \approx 1/(1-\rho)$.

Continuous argmax and discretization. Because price is continuous, the maximization of the UCB index is, in principle, an infinite-dimensional search. In one dimension, this is computationally mild, but it is still useful to be explicit about practical and theoretical choices.

In practice, we compute p_t by evaluating the UCB index on a grid $\mathcal{P}_m \subset [p_\ell, p_h]$ (e.g., evenly spaced or aligned with admissible price endings), selecting the best grid point, and optionally refining with a local optimizer (e.g., Brent search) initialized at that maximizer. This hybrid approach is robust: the grid prevents the optimizer from getting trapped at poor local maxima induced by numerical noise, while local refinement recovers near-continuous performance.

For theory, discretization can be handled in two complementary ways. First, if the posted price must be rounded (as in most retail settings), then the action set is already finite and the algorithm is exactly discrete GP-UCB. Second, if we view discretization as an approximation, we can bound the induced error by controlling the smoothness of the UCB objective. Under our RKHS assumption with squared-exponential kernel, functions are Lipschitz on compact domains, so using a sufficiently fine grid (mesh size shrinking with T) makes the discretization gap negligible relative to the main regret terms we study.

Hyperparameters and stability. Kernel and noise hyperparameters (e.g., length-scale ℓ, signal variance, and λ) are rarely known in pricing applications. A standard empirical Bayes approach is to re-estimate hyperparameters by maximizing the (windowed or discounted) marginal likelihood at each t using only past data. This aligns with operational workflows where models are retrained daily on a rolling sample. However, it also introduces two limitations. First, hyperparameter optimization can be unstable when data are

scarce or when exploration is limited, leading to pathological length-scales (overfitting) and overly narrow confidence bands. Second, our formal guarantees treat the kernel as fixed; plug-in hyperparameters can be interpreted as model misspecification.

A practical compromise is to (i) restrict hyperparameters to a plausible range (e.g., enforce $\ell \in [\ell_{\min}, \ell_{\max}]$), (ii) regularize toward conservative uncertainty (e.g., avoid very small λ), and (iii) tune κ_t to be slightly larger than the nominal theoretical choice to hedge against misspecification. When computational cost is a concern, windowing also helps: naive GP updates scale as $O(W^3)$ per period, so smaller W is not only more adaptive but also materially cheaper; when W must be large, standard approximations (rank-one Cholesky updates, inducing points, random Fourier features) can be layered on without changing the economic logic of forgetting.

These algorithmic details set up the next step: we will formalize how windowing or discounting yields confidence intervals that separate (i) statistical uncertainty from (ii) a drift-induced bias term, and how that separation drives a dynamic regret bound with an explicit bias–variance tradeoff in W (or ρ).

5 Theory I: confidence sets for a windowed GP under drift

The UCB principle is only as good as the confidence set it relies on. In the stationary GP-UCB analysis, one proves that the GP posterior mean $\mu_{t-1}(\cdot)$ concentrates around a *single* unknown function $f(\cdot)$ that generated all past data. In our pricing environment, the object of interest at date t is instead $f_t(\cdot)$, while the observations in the window $\mathcal{D}_{t-1}^{(W)}$ were generated by the time-varying sequence $\{f_s\}_{s=t-W}^{t-1}$. Windowing is therefore not merely a computational device; it is what makes learning meaningful when the target moves. The technical task in this section is to make precise how a windowed GP posterior yields a valid (high-probability) envelope for f_t, and to separate that envelope into (i) a statistical uncertainty term that shrinks with data and (ii) a drift-induced bias term that grows with the window length.

Intuition: treat drift as structured contamination. Fix a date t and consider the observations in the current window. For each $s \in \mathcal{I}_{t-1} = \{\max\{1, t-W\}, \dots, t-1\}$ we observe

$$y_s \;=\; f_s(p_s) + \varepsilon_s \;=\; f_t(p_s) + \underbrace{\big(f_s(p_s) - f_t(p_s)\big)}_{\text{drift mismatch}} + \varepsilon_s.$$

From the perspective of estimating f_t, the data are generated by f_t but with an additional, non-stochastic (and generally non-mean-zero) perturbation

term $f_s(p_s) - f_t(p_s)$. This term is *not* controlled by sub-Gaussian concentration; it is controlled only through the variation budget V_T. Thus, even if the GP posterior based on $\mathcal{D}_{t-1}^{(W)}$ were statistically sharp, it can be systematically biased by the inclusion of stale samples. The role of the window length W is exactly to balance these two forces.

A windowed confidence bound with an explicit drift term. We state a representative concentration result in the form we will use later. The first term is the familiar GP-UCB radius, expressed through the posterior standard deviation and a confidence parameter; the second term is a deterministic drift penalty that captures the worst-case mismatch between f_t and the functions that generated the windowed observations.

Proposition 5.1 (Windowed GP confidence under drift)**.** *Fix a window length W and failure probability $\delta \in (0,1)$. Under (H1)–(H2), there exists a choice of confidence parameter $\beta_{t,W}$ (polylogarithmic in W and $1/\delta$ and linear in B^2 and σ^2) such that, with probability at least $1-\delta$, for all dates $t \in \{1,\dots,T\}$ and all prices $p \in [p_\ell, p_h]$,*

$$\left|f_t(p) - \mu_{t-1}^{(W)}(p)\right| \;\le\; \sqrt{\beta_{t,W}}\,\sigma_{t-1}^{(W)}(p) \;+\; \Delta_{t,W},$$

where the drift-bias term can be taken as

$$\Delta_{t,W} \;:=\; \sum_{s=\max\{1,t-W\}}^{t-1} \sup_{p' \in [p_\ell, p_h]} \left|f_t(p') - f_s(p')\right|.$$

Moreover, by a telescoping argument,

$$\sup_{p'} \left|f_t(p') - f_s(p')\right| \;\le\; \sum_{u=s+1}^{t} \sup_{p'} \left|f_u(p') - f_{u-1}(p')\right|,$$

and hence $\Delta_{t,W}$ is controlled by the local variation over the last W periods, with a crude but useful bound

$$\Delta_{t,W} \;\le\; W \cdot \max_{u \in \{\max\{2,t-W\},\dots,t\}} \sup_{p'} \left|f_u(p') - f_{u-1}(p')\right|.$$

Several remarks clarify what this proposition does (and does not) deliver.

Separation of estimation and nonstationarity. The posterior standard deviation $\sigma_{t-1}^{(W)}(p)$ quantifies how informative the recent price experiments are *about the function that generated those experiments.* When the environment is stationary, that is enough. Here it is not: even perfect knowledge of the past functions $\{f_s\}$ would not identify f_t without a restriction on how quickly the curve moves. The term $\Delta_{t,W}$ is exactly the price of that restriction. It is deterministic conditional on the realized sequence $\{f_t\}$ and depends on W in the opposite direction of statistical uncertainty: larger W generally lowers $\sigma_{t-1}^{(W)}(\cdot)$ but increases $\Delta_{t,W}$.

Why $\Delta_{t,W}$ is the right scale. The variation budget $V_T = \sum_{t=2}^{T} \sup_p |f_t(p) - f_{t-1}(p)|$ is a global constraint, but pricing decisions are local in time. Proposition 5.1 makes this locality explicit: if the recent market is stable, then $\Delta_{t,W}$ is small even when V_T is large due to distant shocks (e.g., a one-off holiday). Conversely, if the market is churning in the last W periods, the bias is unavoidably large, reflecting a genuine identification problem rather than a weakness of the method.

Uniformity over a continuous price domain. The bound is stated for all $p \in [p_\ell, p_h]$ simultaneously, which is what we need to justify a maximization-based policy like UCB. Technically, this uniformity can be obtained using standard GP concentration tools coupled with either (i) a discretization/covering argument (leveraging smoothness implied by the squared-exponential RKHS on a compact interval) or (ii) the information-gain machinery that leads to Γ_W in later regret bounds. The key point is that, in one-dimensional pricing, the complexity penalty is mild: uniform control does not fundamentally change the bias–variance logic.

Discounting as a smooth analogue. For ED-GP-UCB, the same decomposition holds with a weighted analogue of $\Delta_{t,W}$:

$$\Delta_{t,\rho} \;\approx\; \sum_{s=1}^{t-1} \rho^{\,t-s} \sup_{p'} |f_t(p') - f_s(p')|,$$

so that very old observations contribute negligibly. This makes precise the heuristic mapping $W_{\text{eff}} \approx 1/(1-\rho)$: the estimation radius behaves as if we had about W_{eff} effective samples, while the drift bias behaves as if we were comparing f_t primarily to the last W_{eff} periods.

Limitations and what we do with them. Two caveats are worth flagging. First, $\Delta_{t,W}$ is not observable, so the algorithm does not (and cannot) subtract it in real time; instead we choose W (or ρ) to make the worst-case accumulated drift penalty manageable. Second, our confidence parameter $\beta_{t,W}$ treats kernel hyperparameters as fixed; plug-in estimation of ℓ and λ can tighten or loosen the interval in practice, but it lies outside the formal guarantee. These limitations are precisely why the next step is a regret analysis: what matters for performance is not pointwise confidence per se, but how the confidence width and drift bias accumulate along the realized sequence of posted prices. This is the object of our dynamic regret bounds in the next section.

6 Theory II: dynamic regret bounds and how to tune the window

Our goal is not merely to form a pointwise confidence band for f_t, but to translate that band into *economic performance*: how much revenue we lose relative to a seller who knows the entire path $\{f_t\}_{t=1}^T$ and posts the contemporaneous monopoly price $p_t^* \in \arg\max_p f_t(p)$ each period. This is a demanding benchmark—the oracle tracks a moving target—so the relevant question is how the loss scales with two primitives: (i) statistical difficulty (noise and function complexity under k) and (ii) *market instability* as measured by V_T.

Intuition: regret inherits the bias–variance tradeoff. Fix t and suppose SW-GP-UCB chooses p_t by maximizing $\mu_{t-1}^{(W)}(p) + \kappa_t \sigma_{t-1}^{(W)}(p)$. When p_t^* lies inside our confidence envelope, a standard UCB argument bounds the one-step regret $f_t(p_t^*) - f_t(p_t)$ by (a constant multiple of) the confidence radius at the chosen point. Proposition 5.1 tells us that this radius has two parts: a statistical term $\sqrt{\beta_{t,W}}\sigma_{t-1}^{(W)}(p)$ and a drift term $\Delta_{t,W}$. Summing over t then yields two corresponding contributions to cumulative regret: an *estimation* term that decreases with W (more effective samples) and a *tracking* term that increases with W (more staleness). The theorem below formalizes this decomposition.

Theorem 6.1 (Dynamic regret of SW-GP-UCB)**.** *Let SW-GP-UCB select*

$$p_t \in \arg\max_{p\in[p_\ell,p_h]} \mu_{t-1}^{(W)}(p) + \kappa_t \sigma_{t-1}^{(W)}(p), \qquad \kappa_t = \sqrt{\beta_{t,W}},$$

where $\beta_{t,W}$ is chosen so that Proposition 5.1 holds with failure probability δ. Under (H1)–(H3), with probability at least $1-\delta$,

$$\mathrm{Reg}_T^{\mathrm{dyn}} \;=\; \sum_{t=1}^{T} \big(f_t(p_t^*) - f_t(p_t)\big) \;\le\; \tilde{O}\Big(\frac{T}{\sqrt{W}}\sqrt{\Gamma_W} \;+\; V_T W\Big),$$

where Γ_W is the maximum information gain of the GP model over W points on $[p_\ell, p_h]$, and $\tilde{O}(\cdot)$ suppresses polylogarithmic factors in $T, W, 1/\delta$ and constants depending on (B, σ).

Proof sketch (economic reading). The argument mirrors the stationary GP-UCB proof, but with one additional accounting identity. First, by Proposition 5.1, with high probability we have for all p,

$$f_t(p) \le \mu_{t-1}^{(W)}(p) + \sqrt{\beta_{t,W}}\sigma_{t-1}^{(W)}(p) + \Delta_{t,W}, \qquad f_t(p) \ge \mu_{t-1}^{(W)}(p) - \sqrt{\beta_{t,W}}\sigma_{t-1}^{(W)}(p) - \Delta_{t,W}.$$

Evaluating at p_t^* and at the chosen p_t, and using the maximizing property of the UCB rule, yields an instantaneous regret bound of the form

$$f_t(p_t^*) - f_t(p_t) \;\le\; 2\sqrt{\beta_{t,W}}\,\sigma_{t-1}^{(W)}(p_t) \;+\; 2\Delta_{t,W}.$$

Second, summing the posterior standard deviations along the realized actions is controlled by information gain: on any block of length W, the usual GP-UCB analysis gives

$$\sum_{t\in\text{block}} \sigma_{t-1}^{(W)}(p_t) \;\le\; \tilde{O}\big(\sqrt{W\Gamma_W}\big),$$

and a block decomposition over T periods converts this into the global term $\tilde{O}\big(\frac{T}{\sqrt{W}}\sqrt{\Gamma_W}\big)$. Third, the drift term sums to at most order $V_T W$ because each local shock $\sup_p |f_u(p) - f_{u-1}(p)|$ can affect (at most) the next W windows. Economically, each change in market conditions creates a transient period during which the seller is partially "learning yesterday's curve"; the window length determines how long that transient lasts.

A corollary for one-dimensional squared-exponential pricing. In our baseline pricing problem the action space is one-dimensional and compact. For the squared-exponential kernel on $[p_\ell, p_h]$, it is standard that Γ_W grows slowly, e.g.

$\Gamma_W = O(\log^2 W)$ (up to constants depending on the length-scale and domain).

Plugging this into Theorem 6.1 yields the more transparent expression

$$\mathrm{Reg}_T^{\mathrm{dyn}} \;\le\; \tilde{O}\left(\frac{T}{\sqrt{W}}\log W \;+\; V_T W\right).$$

Thus, in one-dimensional pricing, the kernel complexity is not the main driver; the key economic lever is the instability index V_T.

Choosing W: an optimal tracking rate. Treating slowly varying log factors as constants, the upper bound is minimized by balancing the estimation term $T/\sqrt{W}$ against the drift term $V_T W$. This yields the canonical choice

$$W^* \;\asymp\; \left(\frac{T\sqrt{\Gamma_T}}{V_T}\right)^{2/3} \qquad \Rightarrow \qquad \mathrm{Reg}_T^{\mathrm{dyn}} \;\le\; \tilde{O}\big(T^{2/3}V_T^{1/3}\big),$$

with an additional mild $(\Gamma_T)^{1/6}$ factor if we keep track of Γ_W explicitly. The rate $T^{2/3}V_T^{1/3}$ is the familiar "tracking" rate from nonstationary bandits: when the optimum moves, one cannot generally do better than a 2/3 exponent in T without stronger structure. Our contribution here is to show that the same economic logic carries over to a smooth (GP) demand environment with continuous prices.

How the bound behaves across regimes of market instability. Two limiting cases sharpen intuition. If $V_T = 0$ (stationarity), we can take W as large as T and recover the stationary GP-UCB behavior:

$$\mathrm{Reg}_T^{\mathrm{dyn}} \;\le\; \tilde{O}\big(\sqrt{T\Gamma_T}\big),$$

so the seller learns the fixed revenue curve and eventually exploits it. At the other extreme, if V_T is large (rapid churn), the optimal W^* shrinks and the bound approaches linear behavior in T (no algorithm can track an arbitrarily fast-moving optimum). Between these extremes, V_T plays the role of an *economic sufficient statistic* for the value of longer memory: stable markets reward aggregation of data; unstable markets reward myopia.

Practical tuning and limitations. The window rule requires knowledge of V_T to select W^*. In many pricing applications, V_T is not observable ex ante; one can therefore (i) tune W on a coarse grid and update it periodically (a "meta-policy" over windows), or (ii) use exponential discounting with a handful of candidate ρ's to obtain robustness to unknown drift. Either way, the theorem tells us what to look for empirically: performance should be relatively flat near the minimizer of the U-shaped curve $W \mapsto \frac{T}{\sqrt{W}}\sqrt{\Gamma_W} + V_T W$, and the dominant failure mode of overly large W is systematic lag after regime shifts, not statistical noise.

7 Extensions: frictions, structure, and context

The baseline model is intentionally lean: each period we post any price in a compact interval, observe noisy revenue, and update a (windowed) GP posterior while the revenue curve drifts subject to a variation budget. Many pricing environments add constraints that are economically central but do not fundamentally change the learning logic. We briefly discuss three such extensions and how they map into the same bias–variance accounting.

(i) Switching costs and limited price changes via batching. In retail and platform settings, changing prices is not free: there may be menu costs, customer fairness concerns, or operational constraints that make frequent repricing undesirable. One reduced-form approach is to subtract a switching penalty, e.g.

$$\sum_{t=1}^{T} r_t \;-\; c\sum_{t=2}^{T} \mathbf{1}\{p_t \neq p_{t-1}\} \qquad \text{or} \qquad \sum_{t=1}^{T} r_t \;-\; c\sum_{t=2}^{T} |p_t - p_{t-1}|.$$

A convenient way to incorporate such frictions is *batching*: we restrict the seller to update the posted price only every H periods, holding it fixed within each batch. The decision problem then lives on a coarser time index $b =$

$1, \ldots, \lceil T/H \rceil$; within batch b we post a single price $p^{(b)}$, collect the H realized feedback signals, and update the GP posterior once per batch (still using a window of the last W batches, or the last WH raw observations, depending on implementation). Economically, batching translates a switching cost into a hard cap on the number of price moves (at most $\lceil T/H \rceil - 1$), so the total switching penalty is controlled by design.

Batching introduces a new tradeoff. Larger H reduces switching frequency (and thus switching costs), but it also creates *within-batch staleness*: if f_t drifts inside the batch, a price chosen at the batch start is partially targeting yesterday's curve. In regret terms, this adds an approximation component on the order of the cumulative variation occurring within each batch. Under bounded variation, a crude bound is proportional to $\sum_b \sum_{t \in b} \sup_p |f_t(p) - f_{t_b}(p)|$, which is at most $O(V_T H)$ in the worst case. Thus, batching makes the effective tracking term worse by a factor tied to the batch length, while it improves the switching-cost objective. In applications, we interpret H as a policy lever reflecting organizational constraints: tighter governance (smaller H) supports faster tracking, while looser governance (larger H) economizes on repricing effort.

A related (and often more realistic) constraint is *limited price movement*, e.g. $|p_t - p_{t-1}| \leq \eta$. This does not require batching, but it does change the action set at each time. Algorithmically, we can implement SW-GP-UCB with a constrained maximization step,

$$p_t \in \arg \max_{p \in [p_\ell, p_h],\, |p - p_{t-1}| \leq \eta} \mu_{t-1}^{(W)}(p) + \kappa_t \sigma_{t-1}^{(W)}(p),$$

so learning proceeds as before while the feasible set becomes local. The economic consequence is intuitive: with an upper bound on adjustment speed, regret can be dominated by *inability to chase* rapid movements of p_t^*, even when statistical uncertainty is small.

(ii) Seasonal drift and periodic variation budgets. Many demand environments are not drifting arbitrarily; instead they exhibit structured seasonality (day-of-week, pay cycles, holidays). A parsimonious decomposition is

$$f_t(p) \;=\; g(p) + s_{\phi(t)}(p), \qquad \phi(t) \in \{1, \ldots, P\},$$

where $\phi(t)$ is a known phase (e.g. weekday) and s_ϕ is periodic with period P. In such settings, the raw variation budget V_T can be misleadingly large if we ignore phase and compare adjacent days with different seasonal components; yet the environment is predictable once phase is accounted for. Two implications follow.

First, even within the sliding-window paradigm, seasonality suggests a principled window choice: taking W on the order of one period P reduces systematic mismatch, because the window then contains comparable phases.

This is a simple operational heuristic: "learn from last week, not from yesterday" when yesterday is the wrong phase.

Second, if phase is observable, we can *encode* seasonality directly by running a GP on an augmented input (p, ϕ) with a product kernel $k((p,\phi),(p',\phi')) = k_p(p,p')\,k_\phi(\phi,\phi')$, where k_ϕ is periodic (or simply a kernel on the discrete set of phases). In the idealized case where the mapping $(p,\phi) \mapsto f(p,\phi)$ is stationary over t, the problem becomes a contextual but *non-drifting* GP bandit, and the performance guarantee reverts toward stationary GP-UCB rates (with information gain now computed on the augmented domain). Practically, this approach reduces the "effective" drift the algorithm must track: what appears as instability in calendar time becomes stable structure in phase time. When seasonality is only approximate (e.g. holiday effects shift year to year), a hybrid approach—phase-aware kernels combined with discounting or windowing—can deliver robustness.

(iii) Contextual drift: adding covariates x_t. A second form of structure comes from observed covariates that shift demand, such as traffic, inventory visibility, competitor prices, or ad spend. Let $x_t \in \mathcal{X}$ be observed before pricing at time t, and suppose expected revenue is $f_t(p) = F_t(p, x_t)$. A natural extension is to treat pricing as a *contextual* GP-UCB problem on the joint space $[p_\ell, p_h] \times \mathcal{X}$, selecting

$$p_t \in \arg\max_p \; \mu_{t-1}^{(W)}(p, x_t) + \kappa_t \sigma_{t-1}^{(W)}(p, x_t),$$

where the posterior is formed from recent tuples (p_τ, x_τ, y_τ). The same decomposition that drives the dynamic regret bound continues to apply, now with an information-gain term appropriate to the higher-dimensional input and a drift term that measures variation of $F_t(\cdot,\cdot)$ over time.

The main economic message is that covariates can substitute for memory. If a substantial portion of what looks like "drift" is in fact explained by x_t (say, weekends or competitor promotions), then conditioning on x_t reduces the unexplained variation budget and allows the seller to use longer effective windows without staleness. The limitation is equally clear: richer contexts increase statistical complexity, potentially inflating information gain and slowing learning unless we impose additional structure (low-dimensional $\mathcal{X}$, additive kernels, or strong smoothness across x). Finally, in some applications covariates are not exogenous (e.g. traffic can respond to price); then the observed x_t is partly an outcome of the seller's action, and the interpretation shifts from pure prediction to causal learning, requiring either experimentation design or instrumental variation beyond the scope of the present framework.

8 Experiments: tracking under synthetic drift and a repricing-bot stylization

We complement the theoretical guarantees with simulations designed to isolate the economic forces emphasized by the bound: statistical uncertainty (captured by posterior variance and information gain) versus staleness from nonstationarity (captured by the variation budget). Because field data typically confound demand shocks, seasonality, and strategic behavior, we begin with controlled "ground-truth" environments where we can compute the dynamic oracle benchmark and vary the drift pattern holding noise fixed.

Design and evaluation metrics. We fix a feasible price interval $[p_\ell, p_h]$ and generate a time-indexed expected revenue curve $f_t(\cdot)$ satisfying the RKHS boundedness assumption by constructing f_t as a smooth function (a sum of a few squared-exponential bumps) and then letting its parameters move over time. At each t the algorithm posts p_t and observes $y_t = f_t(p_t)+\varepsilon_t$ with ε_t i.i.d. σ-sub-Gaussian (Gaussian in the simulations). We evaluate (i) dynamic regret $\sum_{t=1}^{T}(f_t(p_t^*) - f_t(p_t))$, where p_t^* is computed by dense-grid maximization of f_t; and (ii) average realized revenue as a fraction of the oracle revenue. When we include switching frictions in the objective, we also report the number and magnitude of price changes.

Drift scenarios. We consider three canonical nonstationarity patterns that span many operational settings. *Sinusoidal drift* captures smooth seasonality and gradual macro shifts: we let the argmax of f_t move continuously (e.g. by shifting the center of a unimodal revenue curve) with a known period P, so that V_T scales linearly in T/P for fixed amplitude. *Piecewise-constant drift* captures regime changes such as competitor entry, promotion cycles, or product-page redesigns: we draw a sequence of stationary curves and switch abruptly every L periods, creating concentrated variation at change points. *Adversarial bounded-variation drift* stresses the model class: we let an adaptive procedure choose f_t to be difficult for the learner, subject only to a budget constraint $\sum_{t=2}^{T}\sup_p |f_t(p) - f_{t-1}(p)| \le V_T$; operationally, we implement this by moving the location of the revenue maximizer in a way that "chases" the learner while respecting a step-size cap implied by V_T.

Algorithms and baselines. Our main methods are SW-GP-UCB (a hard window of length W) and ED-GP-UCB (exponential discounting with factor ρ). We compare against three classes of benchmarks. First, *stationary Bayesian optimization* baselines that ignore drift: a standard GP-UCB/BO rule fit on all past data (effectively $W = t-1$) and a strong stationary method (BO-Inf-style optimization) that is competitive when the objective is time-invariant but has no mechanism to forget stale data. Second, *reinforcement*

learning baselines: PPO and SAC with continuous actions, trained online with a replay buffer. To make the comparison meaningful under drift, we also test variants with a finite replay buffer (a crude analog of windowing) and with higher learning rates (to encourage rapid adaptation). Third, *simple heuristics* that practitioners often deploy: (a) fixed price chosen from an initial exploration phase; (b) ϵ-greedy over a discretized price grid; and (c) a "hill-climbing" rule that perturbs the last price up or down based on recent revenue changes.

Main findings: the bias–variance tradeoff appears sharply in practice. Across all drift scenarios, methods with explicit forgetting dominate stationary baselines once drift is nontrivial. In sinusoidal environments, SW-GP-UCB and ED-GP-UCB track the moving maximizer closely; the stationary BO baselines tend to "average" across phases and converge toward a compromise price that is rarely optimal at any particular time. In piecewise-constant environments, the windowed and discounted methods show short-lived regret spikes immediately after a regime change and then re-learn quickly, while stationary BO exhibits persistent post-change bias because pre-change observations continue to pull the posterior mean toward the old curve. In adversarial bounded-variation environments, all methods deteriorate, but SW/ED variants retain a clear advantage: the performance gap is driven less by optimization quality (UCB is not the bottleneck) and more by the ability to prevent obsolete samples from dominating inference.

The RL baselines illustrate a complementary limitation. With careful tuning and sufficiently long horizons, PPO/SAC can eventually learn good pricing policies in slowly drifting settings, but they are markedly less sample-efficient: they require substantially more interaction to match the revenue achieved by GP-based methods, and their performance is sensitive to learning-rate and replay-buffer choices. Under abrupt regime switches, RL agents often overfit to earlier regimes unless the replay buffer is aggressively truncated, in which case variance increases and training becomes unstable. This pattern is consistent with the economic interpretation that policy-gradient methods are powerful function approximators but do not, by default, implement the explicit staleness control that our regret decomposition isolates.

Sensitivity to W and ρ: U-shaped curves and practical tuning. Varying the window length W produces the predicted U-shaped relationship between performance and memory. Small W yields noisy posteriors and excessive exploration (high variance), while large W yields biased estimates after drift (high staleness). The minimizing W shifts systematically with the drift rate: faster drift (larger V_T induced by shorter periods or more frequent regime changes) favors smaller windows. Discounting exhibits the

same pattern when we map ρ to an effective memory $W_{\text{eff}} \approx 1/(1-\rho)$: larger ρ helps in slowly drifting environments but hurts when changes are abrupt. Empirically, ED-GP-UCB tends to be more forgiving than hard windowing when drift is smooth (sinusoidal), because gradual downweighting avoids sharp posterior discontinuities; conversely, strict windowing can react slightly faster to regime breaks.

From a deployment perspective, these sensitivity patterns suggest a simple operational rule: tune W (or ρ) to the characteristic timescale of change that the business can tolerate. When only rough prior knowledge is available, we find that lightweight online tuning—e.g. selecting W from a small candidate set using recent held-out predictive likelihood or recent revenue—captures much of the benefit without requiring direct estimation of V_T. The broader takeaway is that the window/discount parameter is not a mere technicality: it is the algorithmic representation of an economic stance on how quickly the environment is believed to move, and how costly it is to respond to that movement.

9 Discussion and limitations

Our theory and simulations highlight a simple message: in nonstationary pricing, *forgetting is an economic design choice*. The window length W (or discount factor ρ) is the algorithmic representation of how quickly we believe the revenue landscape moves and how much we are willing to pay, in foregone revenue, to keep tracking it. That said, the guarantees rely on modeling decisions—especially the kernel prior, the observation model, and the bounded-variation assumption—that can be violated in practice. We discuss the main failure modes and what they imply for deployment.

Kernel misspecification and "smoothness risk." We work with a squared-exponential kernel, which encodes very strong smoothness: sample paths are infinitely differentiable, and the associated information-gain term Γ_W is benign (polylogarithmic in one dimension). This is analytically convenient and often empirically reasonable when price is a one-dimensional action, but it can be optimistic when the true mapping $p \mapsto f_t(p)$ has sharp kinks (e.g. threshold effects from shipping cutoffs, discrete competitor price matching, or psychological pricing around .99) or when the revenue curve is multimodal in a way that violates the implied length-scale. In such cases, the GP posterior can be systematically overconfident away from observed prices, and the UCB rule may under-explore the regions where the model is wrong. The regret bounds do not directly apply when $f_t \notin \mathcal{H}_k$, and in finite samples the resulting error can be economically meaningful: the algorithm may "lock in" to a locally good but globally suboptimal price because the kernel extrapolates too aggressively.

Operationally, we view kernel choice less as a statistical nicety and more as a statement about demand microfoundations. If management believes that small price changes should not radically alter conversion, a smooth kernel is defensible; if institutional constraints (rounding, stepwise shipping fees, discrete platform rules) create discontinuities, then Matérn kernels with lower smoothness, additive or piecewise kernels, or even mixtures that allow both smooth and localized components can be safer. A pragmatic compromise is to maintain a small library of kernels and select among them by rolling predictive likelihood within the same windowing/discounting scheme, so that model selection itself does not over-weight stale regimes.

Hyperparameter drift and endogenous uncertainty. Even if the kernel family is appropriate, its hyperparameters (length-scale, signal variance, noise scale) may change over time. In pricing, this is not exotic: a redesign of the product page can make demand less noisy; a new competitor can make the revenue curve sharper; an advertising campaign can change the relevant scale over which price matters. If we fit hyperparameters once and hold them fixed, the posterior variance $\sigma_{t-1}^{(W)}(\cdot)$ may become miscalibrated precisely when the environment changes, undermining the exploration term that drives both learning and the confidence arguments behind UCB.

Allowing online hyperparameter estimation creates a second tension. Fast re-estimation improves adaptivity but can induce feedback loops: the algorithm chooses prices based on a posterior computed from the same data used to tune the model, and aggressive tuning can shrink uncertainty in a self-confirming way. In practice, we recommend treating hyperparameter updates as a controlled process: re-estimate on a slower clock than price updates; regularize toward conservative length-scales (shorter ℓ tends to reduce risky extrapolation); and stress-test calibration by checking empirical coverage of residuals within the sliding window. From an economic perspective, this is akin to governance over the firm's "measurement system": when the market regime changes, we should expect both the optimal price *and* the reliability of our measurement to change.

Partial observability: censored demand, covariates, and strategic data. Our baseline observation model $y_t = f_t(p_t) + \varepsilon_t$ abstracts from several data realities. First, in finite inventory, sales are censored: we observe $q_t = \min\{s_t, D_t(p_t)\}$, so revenue understates demand when stockouts occur. Treating r_t as unbiased feedback can then bias the GP downward in high-demand states, pushing prices in the wrong direction. Second, many firms observe rich covariates (traffic, ad spend, macro indicators) and care about context-dependent pricing; omitting these covariates can inflate the apparent drift V_T because the algorithm attributes predictable variation to time. Third, the data-generating process may be strategic: consumers and com-

petitors can react to pricing policies, making $D_t(\cdot)$ policy-dependent rather than an exogenous stochastic function. Our framework is best interpreted as capturing the reduced form faced by a seller who treats the environment as drifting but not strategically responding to the learning rule.

These issues are not merely technical. They shape what the algorithm is *allowed* to learn from, and thus what forms of experimentation are safe. Where censoring is severe, it may be preferable to learn from upstream signals (e.g. conversions, add-to-cart, or lost-sales estimates) rather than revenue alone; where covariates are available, a contextual GP over (p, x_t) can reduce "effective" nonstationarity by explaining away predictable shifts; and where strategic interaction is central, regret relative to a nonstrategic dynamic oracle may be the wrong benchmark.

When bounded-variation drift is violated. The variation budget V_T is a disciplined way to model change, but it does rule out some economically relevant shocks. Flash crashes in demand, one-time policy interventions, and discontinuous platform changes can generate large instantaneous jumps that dominate $\sum_t \sup_p |f_t - f_{t-1}|$. When such events occur, the U-shaped tradeoff in W becomes more acute: long memory is harmful, but extremely short memory makes the posterior noisy and can lead to costly price oscillations. In these settings, hybrid procedures that combine forgetting with explicit *change detection* (e.g. monitoring predictive errors and triggering a reset) are often more robust than any fixed W or ρ. More broadly, bounded-variation is a modeling stance that says "the world moves, but not arbitrarily fast"; if the business environment does not respect that stance, then no purely passive tracking method can guarantee low regret without additional structure or side information.

Policy and operational implications. Two deployment implications follow. First, choosing W (or ρ) should be tied to an interpretable timescale: how quickly do we believe willingness-to-pay, competitive conditions, or traffic composition changes in a way that matters for pricing? This suggests governance: teams can set a default memory horizon (say, "two weeks of evidence"), then allow limited online adaptation around it, rather than treating tuning as a purely algorithmic exercise. Second, experimentation has externalities. Even if SW-GP-UCB improves revenue, frequent price moves can erode trust, trigger platform scrutiny, or violate internal fairness policies. These costs can be modeled as switching frictions or explicit constraints on price changes; empirically, we find that such constraints often shift the optimal W upward because the firm is effectively committing to smoother paths. Finally, in regulated or high-salience categories, "safe exploration" matters: conservative uncertainty calibration and explicit caps on price excursions can be as important as asymptotic regret. In short, the methods here are best

seen as components of a pricing system with institutional constraints, not as fully autonomous repricing agents.

Conclusion: practical takeaways and open problems. We take away three practical lessons from the nonstationary GP-UCB perspective. First, the central object to govern is not the point estimate of the revenue curve, but the *effective memory* of the system. Windowing or discounting determines which past regimes are treated as relevant evidence today, and this choice should be expressed in business time units (days, weeks, seasons) rather than only as a tuning parameter. Second, the decision rule must be paired with diagnostics. In drifting environments, the most common deployment failure is not that the algorithm explores too little in a stationary world, but that it continues to trust a model trained on a world that no longer exists. Rolling measures of predictive fit, residual calibration, and the realized frequency of near-boundary prices provide early warnings that the memory horizon is too long, the kernel is too smooth, or the uncertainty is miscalibrated. Third, the economic costs of experimentation are multi-dimensional: short-run revenue loss is only one component, alongside customer trust, operational complexity, and compliance. These costs should be reflected explicitly—as constraints on price moves, caps on exploration ranges, or switching frictions—rather than treated as informal "guardrails" outside the model.

From an implementation standpoint, a useful workflow is to treat SW/ED-GP-UCB as a *tracking controller* with two knobs: a memory knob (W or ρ) and a conservativeness knob (the exploration multiplier and any safety constraints). We can choose memory by aligning it with a presumed "half-life" of demand predictability, then refine it empirically by backtesting rolling regret proxies (e.g. the gap between posted revenue and a hindsight best-in-window price). We can choose conservativeness by specifying acceptable tail risk: how often are we willing to post prices that are meaningfully outside the historically profitable range? In many retail settings, the dominant risk is not that we fail to find the global optimum in a smooth curve, but that we generate unstable price paths that create organizational pushback. In that sense, constraints that smooth prices can be complementary to forgetting: they reduce the harm from noisy short windows, allowing faster tracking without visible oscillations.

Our results also suggest a more nuanced view of "adaptive pricing" as an organizational capability. In stable periods, the algorithm should behave like a high-precision estimator with a long memory; in turbulent periods, it should behave like a change-responsive tracker. A purely fixed W or ρ can only approximate this, which motivates hybrid designs: meta-rules that adapt memory based on forecast errors; periodic "re-anchoring" to a baseline price policy; and model ensembles that hedge across kernels and timescales. Such designs are not merely engineering tricks: they correspond to econom-

ically interpretable commitments about how much the firm is willing to rely on the past, and how quickly it can credibly pivot when the market moves.

Several open problems are especially salient for the economics of nonstationary pricing. The first is *multi-agent competition.* In many categories, the seller's demand is influenced by competitors' prices and promotions, which themselves respond to the seller's policy. Then $f_t(\cdot)$ is not an exogenous drifting function but part of an evolving game. Regret relative to a dynamic oracle can be misleading because the benchmark ignores strategic reactions. The right object may be a notion of regret to a time-varying equilibrium, or performance guarantees under a class of competitor response models (e.g. bounded rationality or slow adaptation). Methodologically, this pushes us toward multi-agent bandits, learning in games, and models where the "environment drift" is endogenous. Practically, it implies that learning systems should be evaluated not only on immediate revenue lift, but also on their impact on the competitive dynamics they may induce.

The second open problem is *fairness and policy constraints.* Modern pricing systems are increasingly subject to constraints: limits on price discrimination, requirements of price transparency, caps tied to cost-based rules, and internal policies aimed at avoiding extreme price dispersion across regions or consumer groups. These constraints interact with exploration in nontrivial ways. A rule that is "fair" in outcomes (e.g. bounded differences in realized prices across groups) may be incompatible with efficient learning if some groups provide more informative signals; conversely, a rule that is fair in opportunity (e.g. equal exploration budgets across groups) may incur avoidable revenue losses. This motivates constrained and multi-objective formulations in which the algorithm learns under parity constraints, Lipschitz constraints across groups, or welfare-based objectives that trade off revenue and consumer surplus. The analytical challenge is to obtain regret guarantees that incorporate both drift and constraints without collapsing into vacuous worst-case bounds; the practical challenge is to produce auditable policies with clear documentation of where and why the system explores.

A third open problem is *finite inventory and operational integration.* When inventory is limited, the feedback is censored and the objective is no longer period-by-period revenue but an intertemporal tradeoff between margin and stock depletion. The seller's decision becomes a joint control of price and inventory, potentially with replenishment and lead times, and the "oracle" policy depends on the remaining stock and the future demand trajectory. Extending the GP tracking approach requires combining nonstationary demand learning with dynamic programming or approximate control, and handling the fact that learning changes the future state distribution (because price affects sales and hence future inventory). A promising direction is to treat the continuation value of inventory as an endogenous "shadow cost" that converts the problem back into a sequence of myopic pricing decisions with an adjusted objective; another is to use contextual kernels where the

state (inventory, time-to-replenish, season) enters as a covariate, reducing apparent drift by explaining it through state dependence.

More broadly, the theory invites work on *robustness* and *evaluation*. Robustness asks how to design tracking algorithms that degrade gracefully under misspecification, heavy-tailed noise, or abrupt regime shifts, without requiring ad hoc resets. Evaluation asks how to assess nonstationary pricing systems with credible counterfactuals, given that the policy changes the data it sees. Both questions matter for practice: firms need not only performance, but also stable governance and clear accountability. Our view is that the main contribution of the framework is to make the tradeoff explicit: memory controls adaptivity, uncertainty controls experimentation, and both should be chosen with the economic institution in mind.