Nonparametric Learning and Earning with One-Point Feedback under Nonstationarity

Xiangyu Yang, Feng Xu, Jian-Qiang Hu, Jiaqiao Hu

fxu25@m.fudan.edu.cn

Shandong University, Fudan University, Stony Brook University

June 2026

Talk Roadmap

1. Core challenge
One price, one revenue signal, changing demand.

2. Model
Dynamic regret and path variation.

3. Algorithms
Mirror ascent, restarting, and meta-learning.

4. Guarantees
Regret rates separating learning and adaptation.

5. Experiments
Ablation, controlled variation, and Walmart-calibrated simulator.

6. Implications
Operational guidance, theoretical interpretation, and open directions.

Example 1: Stochastic Pricing Problem

Linear demand simulator

$$x = (x_1, x_2)^\top \in [-5, 5]^2,$$
$$d_{i,t}(x) = b_{i,t} - x_i/2,$$
$$r_t(x) = x_1 d_{1,t}(x) + x_2 d_{2,t}(x) = b_t^\top x - \|x\|^2/2.$$

- b_t is the moving market condition; $x_t^* = b_t$.
- Feedback is one noisy revenue sample at the posted price.

Core tension

Learn from one stochastic payoff while the best price moves over time.

Why this example is useful

The quadratic case makes the decomposition transparent: regret comes from tracking a moving b_t plus experimentation under one-point feedback.

Example 2: Online Newsvendor Learning

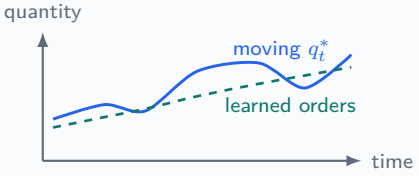
Sequential inventory decisions

$$q_t \in [0, Q] \quad \text{is the order quantity,}$$
$$\Pi_t(q_t) = p \min\{q_t, D_t\} - c q_t - s(q_t - D_t)^+,$$
$$q_t^* = F_t^{-1}\left(\frac{p-c}{p+s}\right).$$

- Demand distribution F_t changes with seasonality, promotions, and shocks.
- The oracle quantity q_t^* moves as the critical fractile changes.
- Feedback: one realized profit sample.

Nonstationary demand

A replenishment policy must relearn the demand quantile while avoiding costly overstock and stockout losses.



Same learning logic: one action, one noisy payoff, and a drifting oracle decision.

Dynamic Pricing Is a Learning Problem

Common structure behind both examples

- Choose one action $\tilde{x}_t \in \mathcal{K}$ before demand is known.
- Observe one noisy payoff $\phi_t(\tilde{x}_t)$.
- Update future actions from past observations only.

Pricing example

action = price; payoff = realized revenue; oracle = best current price.

Newsvendor example

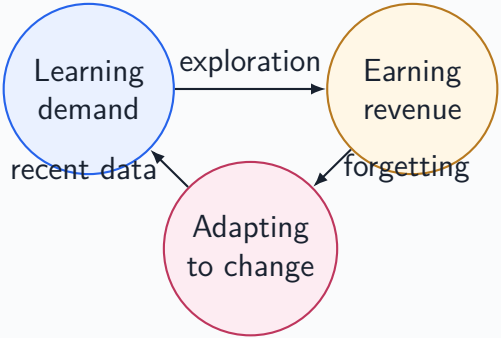
action = order quantity; payoff = realized profit; oracle = current critical-fractile quantity.

$$\phi_t(\tilde{x}_t) = r_t(\tilde{x}_t) + \xi_t, \qquad x_t^* \in \arg\max_{x \in \mathcal{K}} r_t(x).$$

Variation budget definition: total movement of the oracle price path

$$V_T := \max_{\{x_t^* \in \Theta_t^*\}_{t=1}^T} \sum_{t=2}^{T} \|x_t^* - x_{t-1}^*\|, \qquad \Theta_t^* = \arg\max_{x \in \mathcal{K}} r_t(x).$$

Nonstationarity Adds a Third Force



Market conditions can drift or jump due to tastes, competitors, promotions, macro shocks, or product life cycles. Historical demand estimates may lose relevance.

The Information Constraint

Target quantity

The gradient of expected revenue at the current price:

$$\nabla r_t(x_t).$$

Available feedback

A single noisy realized revenue:

$$\phi_t(\tilde{x}_t) = r_t(\tilde{x}_t) + \xi_t.$$

- Multi-point finite differences are natural in stationary settings.
- Under nonstationarity, two revenue queries at two times may come from two different functions.
- The algorithm must learn from one function value per period.

Three Levels of Algorithm Ideas

Level 1: Learn from one point

Perturb the posted price, observe one revenue value, form a gradient proxy, and update by mirror ascent.

Level 2: Forget stale data

Restart the base learner in batches so the policy can track a moving revenue maximizer when V_T is known.

Level 3: Adapt the memory length

Run many restart schedules in parallel and use expert weights to hedge when V_T is unknown.

Algorithm 1 solves sparse feedback; Algorithm 2 adds controlled forgetting; Algorithm 3 removes the need to choose the variation budget in advance.

Pricing Environment

- Horizon: $t = 1, \ldots, T$.
- Price vector: $\tilde{x}_t \in \mathcal{K} \subset \mathbb{R}_+^d$.
- Expected revenue function in period t: $r_t(\cdot)$.
- Observed revenue feedback:

$$\phi_t(\tilde{x}_t) = r_t(\tilde{x}_t) + \xi_t.$$

- Period-wise oracle price:

$$x_t^* \in \arg\max_{x \in \mathcal{K}} r_t(x).$$

The seller does not know the analytical form of r_t and cannot observe gradients or full demand curves.

Regularity: Flexible but Learnable

Revenue functions

For each t, r_t is concave, continuously differentiable, and has L-Lipschitz gradients on $\mathcal{K}$:

$$\|\nabla r_t(x) - \nabla r_t(y)\| \leq L\|x - y\|.$$

- The maximizer lies in an interior convex body $\Theta \subset \mathcal{K}^\circ$.
- Concavity gives tractable local improvement.
- Smoothness controls the cost of perturbing prices.

Performance Metric: Dynamic Regret

Dynamic regret

$$\text{D-Reg}(T) = \sum_{t=1}^{T} r_t(x_t^*) - \mathsf{E}\left[\sum_{t=1}^{T} r_t(\tilde{x}_t)\right].$$

Benchmark. A clairvoyant seller who knows every period's revenue landscape.

Policy. A nonanticipating seller using only past prices and revenues.

Why control change?

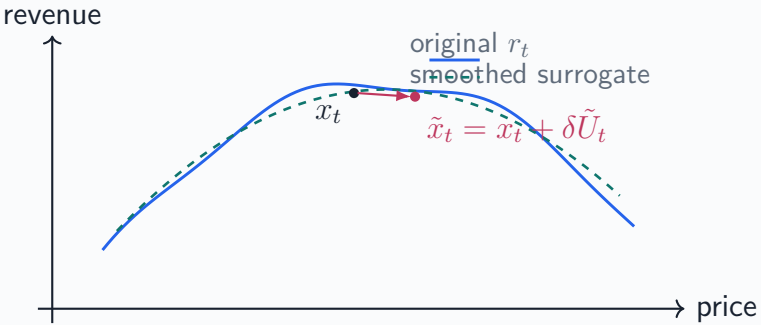
Without a restriction on the oracle path, sublinear dynamic regret is impossible: the best price can move too quickly to track.

Path variation budget

$$\max_{\{x_t^* \in \Theta_t^*\}} \sum_{t=2}^{T} \|x_t^* - x_{t-1}^*\| \le V_T.$$

Small V_T: stable market, longer memory.
Large V_T: volatile market, faster forgetting.

Step 1: Smooth Through Random Perturbation



One-point feedback estimates the gradient of a locally smoothed revenue surface by perturbing the posted price and scaling the observed revenue.

Generic One-Point Gradient Estimator

Estimator

$$G_t = \frac{\phi_t(x_t + \delta \tilde{U}_t)}{\delta} \bar{U}_t.$$

δ
Perturbation radius.

$\tilde{U}_t$
Random price perturbation.

$\bar{U}_t$
Direction used to form the gradient signal.

Covers spherical smoothing, Gaussian smoothing, coordinate perturbations, and simultaneous perturbation schemes.

The Bias–Variance Abstraction

Oracle condition

There exist constants $C_1, C_2, p, q > 0$ such that

$$\|\nabla r_t(x_t) - \mathsf{E}[G_t \mid \mathcal{F}_t]\| \le C_1 \delta^p, \qquad \mathsf{E}\big[\|G_t - \mathsf{E}[G_t \mid \mathcal{F}_t]\|^2 \mid \mathcal{F}_t\big] \le C_2 \delta^{-q}.$$

- Larger δ: more smoothing bias.
- Smaller δ: larger variance after division by δ.
- The analysis only needs this trade-off, not a specific estimator.

Online Mirror Ascent

Update

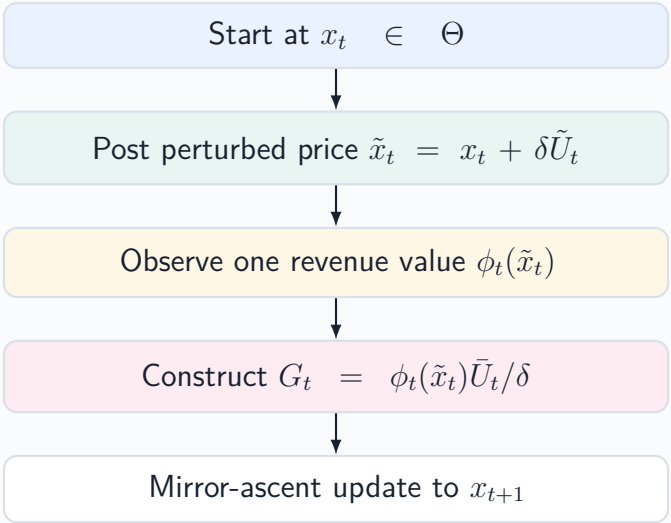
$$x_{t+1} = \arg\max_{x \in \Theta} \left\{ \eta G_t^\top x - B_\psi(x, x_t) \right\},$$

where

$$B_\psi(x, y) = \psi(x) - \psi(y) - \nabla\psi(y)^\top (x - y).$$

- Move in the estimated ascent direction G_t.
- Penalize unstable movement through Bregman distance.
- With $\psi(x) = \|x\|^2/2$, this becomes projected gradient ascent.

Algorithm 1: Base Learner



Static Regret Bound

Proposition

For horizon τ, the base learner satisfies

$$\text{S-Reg}(\tau) \leq \frac{B}{\eta} + \frac{\tau\eta}{\alpha}\left(C_1^2\delta^{2p} + \frac{C_2}{2}\delta^{-q} + C_r\right) + \tau C_1\delta^p\sqrt{d} + \frac{\tau L C_u}{2}\delta^2.$$

initial uncertainty | estimation variance | smoothing bias | perturbation loss

Optimizing the Base Learner

Dominant trade-off

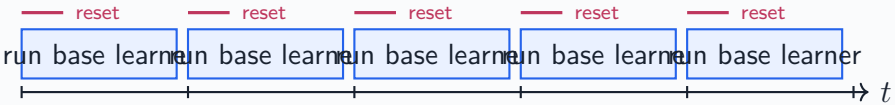
$$\frac{B}{\eta} + \frac{\tau \eta C_2}{2\alpha} \delta^{-q} + \tau \widehat{C} \delta^{\widehat{p}}, \qquad \widehat{p} = \min\{p, 2\}.$$

Choosing η and δ to balance optimization, variance, and smoothing gives

$$\text{S-Reg}(\tau) = O\left(\tau^{(\widehat{p}+q)/(2\widehat{p}+q)}\right).$$

For common spherical smoothing with $p = q = 2$, this is $O(\tau^{2/3})$.

From Static to Dynamic: Restarting



Batch size τ controls how quickly older observations are discounted.

- Longer batches: better learning within each stable window.
- Shorter batches: faster adaptation to moving optima.

Algorithm 2: Restarting Mirror Ascent

Inside each batch

Run Algorithm 1 with batch-specific δ_b, η_b.

At a restart

Discard the current state and initialize a fresh mirror-ascent instance.

Restart scale

The batch size τ should reflect T and V_T.

Stable market $\Rightarrow$ large τ.
Volatile market $\Rightarrow$ small τ.

Dynamic Regret Decomposition

Theorem

The restarting policy satisfies

$$\sum_{t=1}^{T} r_t(x_t^*) - \mathsf{E}\left[\sum_{t=1}^{T} r_t(\tilde{x}_t)\right] \leq \sum_{b=1}^{\lceil T/\tau \rceil} U_b + L_r \tau V_T.$$

$\sum_b U_b$
Cost of learning from noisy one-point feedback inside batches.

$L_r \tau V_T$
Cost of using a static benchmark while the optimum moves.

Known Variation Budget: Optimal Restart Scale

Batch size

$$\tau = \left\lceil \left(\frac{\sqrt{d}\, T}{V_T} \right)^{(2\widehat{p}+q)/(3\widehat{p}+q)} \right\rceil .$$

Dynamic regret

$$\text{D-Reg}(T) = O\Big(\text{poly}(d)\, T^{(2\widehat{p}+q)/(3\widehat{p}+q)} V_T^{\widehat{p}/(3\widehat{p}+q)} \Big) .$$

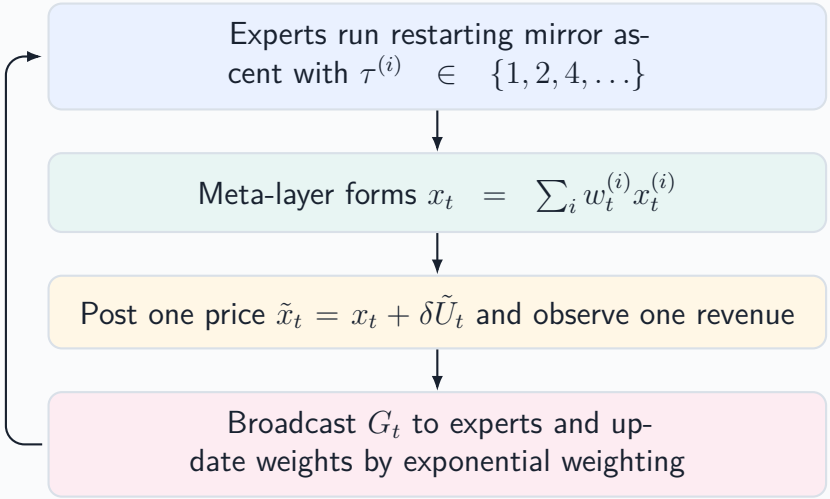
For $p = q = 2$, the time dependence becomes $T^{3/4}$.

Unknown Variation Budget

- Practitioners rarely know a numerical path-variation budget.
- A single restart schedule may underreact during shocks and over-explore during stable periods.
- This motivates a procedure that hedges across multiple memory lengths.

Aggregation strategy. Treat each restart schedule as an expert, aggregate their price recommendations, and update weights using the shared one-point gradient signal.

Algorithm 3: Bandit over Bandit



Expert Weight Update

Exponential weighting

$$w_{t+1}^{(i)} = \frac{w_t^{(i)} \exp\left(\varepsilon G_t^\top x_t^{(i)}\right)}{\sum_{j=1}^{N} w_t^{(j)} \exp\left(\varepsilon G_t^\top x_t^{(j)}\right)}.$$

- Experts do not get separate market experiments.
- All experts share the same realized feedback.
- The meta-layer shifts weight toward restart scales whose recommendations align with profitable ascent directions.

Unknown V_T: Regret Guarantee

Theorem, simplified message

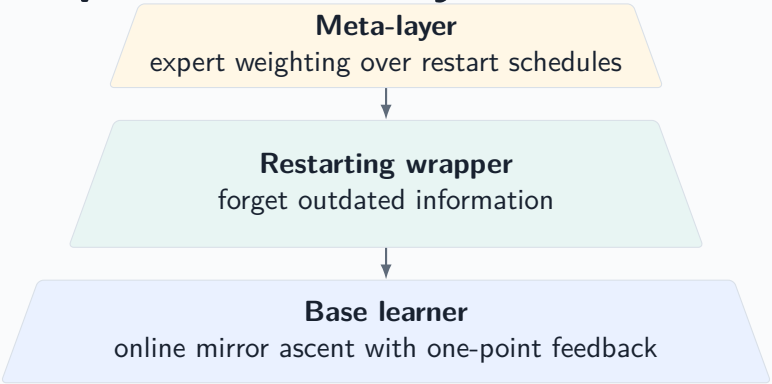
With a geometric expert grid and suitable $\varepsilon, \eta^{(i)}, \delta$,

$$\text{D-Reg}(T) \lesssim T^{2/3} V_T^{1/3} \cdot \text{estimation factor} + \frac{\sqrt{T}}{\delta} \cdot \text{meta cost} + T\delta^{\min\{p,2\}}.$$

Under the standard $p = q = 2$ case and $\delta \asymp T^{-1/9}$, the rate refines to

$$O\left(\text{poly}(d)\, T^{7/9} V_T^{1/3}\right).$$

Conceptual Summary of the Method



The framework separates three responsibilities: estimate a local ascent direction, decide how much memory to keep, and adapt that memory length when variation is unknown.

Reading the Rates

Setting	**Need to know V_T?**	**Typical rate for $p = q = 2$**
Base learner	Not applicable	$\text{S-Reg}(\tau) = O(\tau^{2/3})$
Restarting	Yes	$\text{D-Reg}(T) = O(\text{poly}(d) T^{3/4} V_T^{1/4})$
Expert weighting	No	$\text{D-Reg}(T) = O(\text{poly}(d) T^{7/9} V_T^{1/3})$

The adaptive method pays a modest rate penalty for removing prior knowledge of the variation budget before the numerical evidence begins.

Experimental Program

Ablation
Static learner vs restarting vs meta-learning.

Controlled variation
Synthetic 2D pricing with $V \in \{0, 10, 20, 30, 40\}$.

Real data simulator
Walmart M5 price–sales panel, 54 products.

- Same feedback restriction throughout: one realized revenue per period.
- Primary metric: cumulative dynamic regret.
- Plots report averages across independent replications with confidence intervals where applicable.

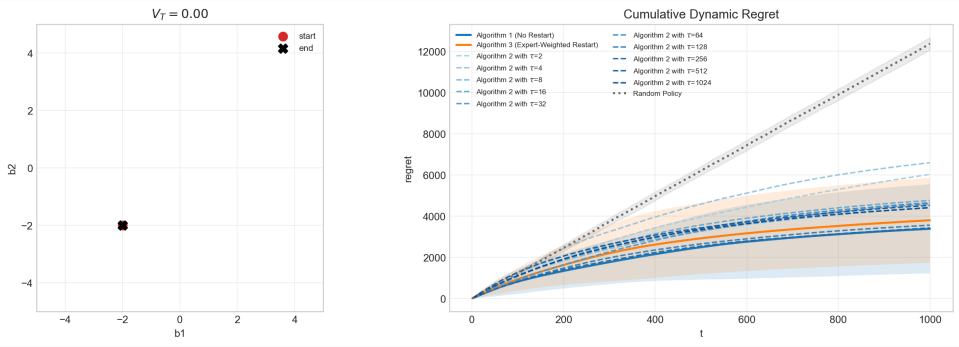
Ablation Setup

Two-product quadratic environment

$$r_t(x) = b_t^\top x - \frac{1}{2}\|x\|^2, \qquad x_t^* = b_t, \qquad x \in [-5, 5]^2.$$

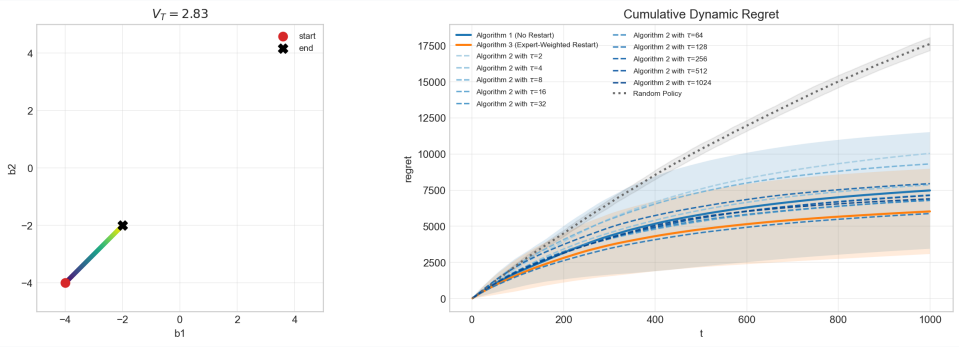
- Compare base mirror ascent, fixed restarting schedules, expert-weighting meta-layer, and random pricing.
- Three regimes: no variation, low variation, high variation.
- Noise: Gaussian revenue perturbation.

Ablation: No Variation



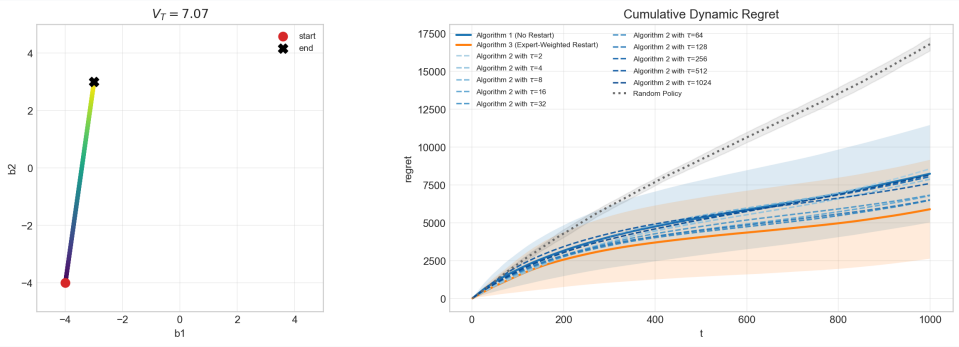
In a stationary environment, restarting is unnecessary and can add learning cost.

Ablation: Low Variation



Mild drift is enough for the advantage of adaptation to begin appearing.

Ablation: High Variation



With faster-moving optima, static learning increasingly lags behind the environment.

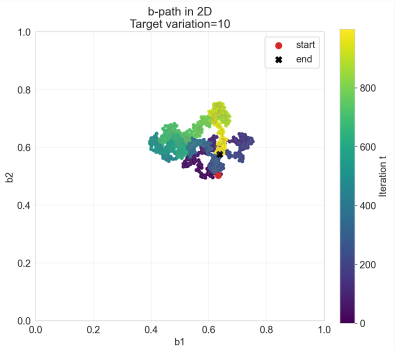
Controlled Variation: How Paths Are Generated

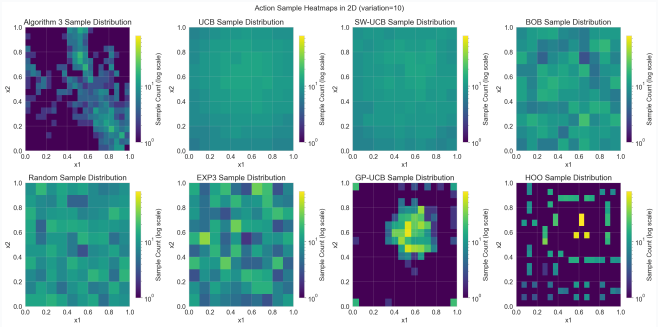
Constrained random walk

$$b_t = \max\{c - r,\, \min\{b_{t-1} + \Delta_V d_t,\, c + r\}\}, \qquad \Delta_V = \frac{V}{T-1}.$$

- $b_t \in [0,1]^2$ controls the optimal price.
- The construction ensures $\sum_{t=2}^{T} \|b_t - b_{t-1}\| \le V$.
- Baselines include UCB, GP-UCB, EXP3, HOO, SW-UCB, BOB, and random pricing.

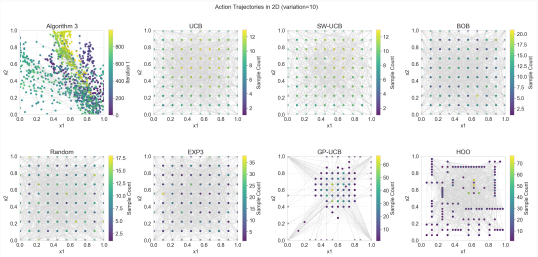
Controlled Variation: $V = 10$ Environment

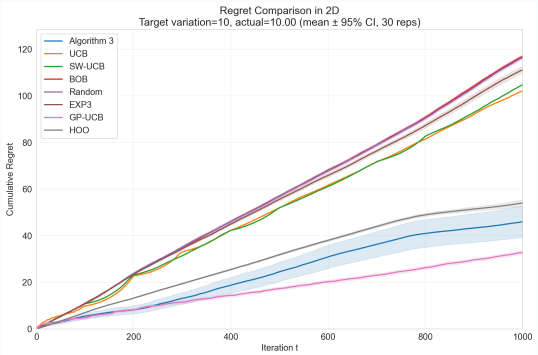




Low variation: the optimal region moves, but tracking remains feasible with moderate memory.

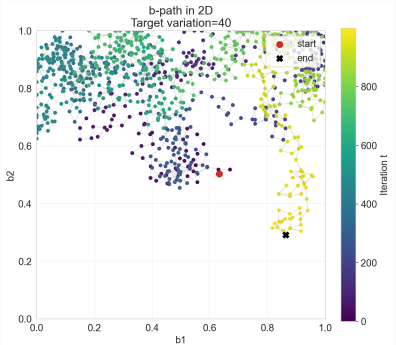
Controlled Variation: $V = 10$ Tracking and Regret

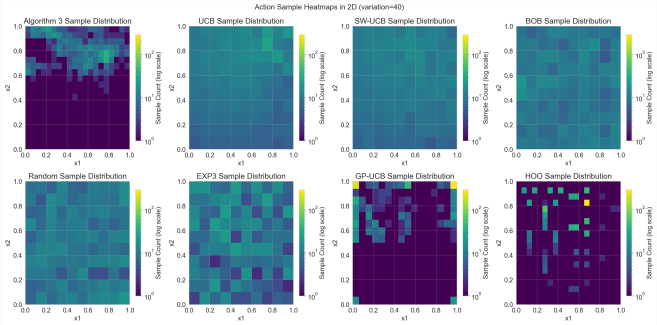




The proposed method samples around the moving high-revenue region and maintains lower regret.

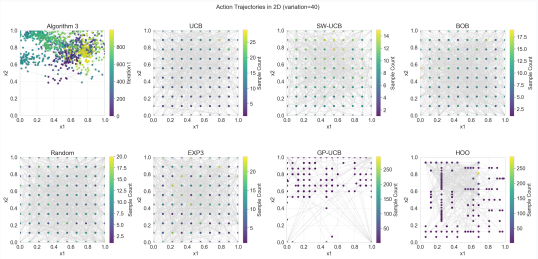
Controlled Variation: $V = 40$ Environment

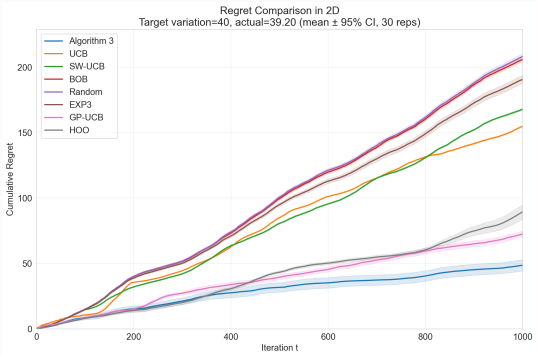




High variation: the environment calls for faster forgetting and more responsive tracking.

Controlled Variation: $V = 40$ Tracking and Regret





The gap widens as nonstationarity increases; discrete baselines incur substantial cost from arm-wise exploration.

Walmart-Calibrated Pricing Simulator

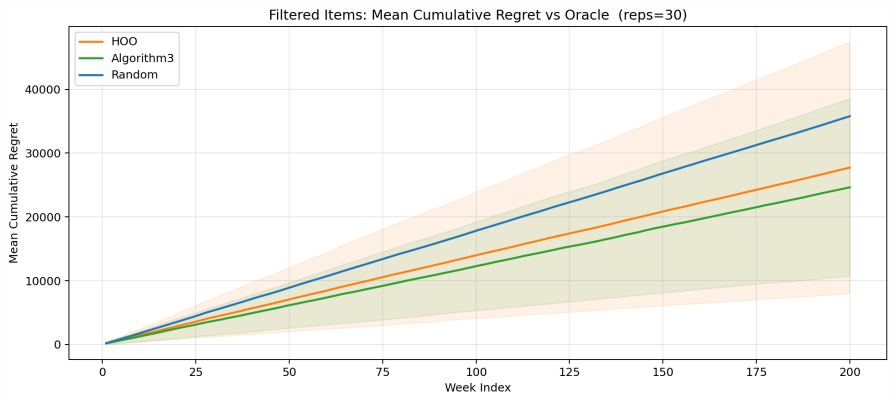
- Walmart M5 daily sales and weekly prices.
- Filter to 54 items with sufficient signal.
- Aggregate daily sales into weekly demand.
- Fit time-varying item-level price–demand curves.

$$D_{i,t} \sim \text{Poisson}(\lambda_{i,t}),$$

$$\lambda_{i,t} = \max\{\hat{a}_{i,t}p_{i,t}^2 + \hat{b}_{i,t}p_{i,t} + \hat{c}_{i,t}, 0\}.$$

$$R_t = \sum_{i=1}^{54} p_{i,t}D_{i,t}.$$

Walmart Experiment: Results



In the high-dimensional calibrated simulator, the expert-weighted restarting method outperforms HOO and random pricing against the oracle benchmark.

Theoretical Implications

One-point feedback
Learning is possible, but smoothing creates a bias–variance trade-off.

Restarting
Dynamic regret is learning cost plus adaptation cost.

Meta-learning
Unknown variation can be handled by hedging across restart schedules.

Operational implication

The algorithm learns prices while automatically adjusting how much historical demand information should influence current decisions.

Implementation Considerations

- Keep a small grid of memory lengths: short, medium, and long.
- In each period, aggregate the price suggestions from those learners.
- Use one observed revenue to update all learners and their weights.
- More volatile markets shift weight toward shorter memory.
- More stable markets shift weight toward longer memory.

The procedure is operationally lightweight: one market action per period, no demand-function specification, and no need to know V_T in advance.

Limitations and Extensions

Operations extensions
Add inventory, capacity, replenishment, or joint price–inventory control under nonstationarity.

Market structure
Consider endogenous nonstationarity from competitors or reference-price effects.

Theory extensions
Sharpen dimension dependence for one-point gradient methods.

Lower bounds
Establish minimax limits under comparable path-variation measures.

Takeaway

Nonstationary pricing with one-point feedback can be addressed when learning, forgetting, and adaptation are designed together.

One-point mirror ascent learns locally; restarting discounts older data; expert weighting selects the appropriate forgetting scale.

Questions and comments are welcome.