Learning to Price: Interpretable Attribute-Level Models for Dynamic Markets

Srividhya Sethuraman
IIT Madras & TCS Research*
Chennai, India
srividhya0197@gmail.com
srividhya.sethuraman1@tcs.com

Chandrashekar Lakshminarayanan
IIT Madras
Chennai, India
chandrashekar@{cse,dsai}.iitm.ac.in

ABSTRACT

Dynamic pricing in high-dimensional markets poses fundamental challenges of scalability, uncertainty, and interpretability. Existing low-rank bandit formulations learn efficiently but rely on latent features that obscure how individual product attributes influence price. We address this by introducing an interpretable Additive Feature Decomposition-based Low-Dimensional Demand (**AFDLD**) model, where product prices are expressed as the sum of attribute-level contributions and substitution effects are explicitly modeled. Building on this structure, we propose **ADEPT** (Additive DEcomposition for Pricing with cross-elasticity and Time-adaptive learning), a projection-free, gradient-free online learning algorithm that operates directly in attribute space and achieves sublinear regret of $\tilde{O}(\sqrt{d}, T^{3/4})$. Through controlled synthetic studies and real-world datasets, we show that ADEPT (i) learns near-optimal prices under dynamic market conditions, (ii) adapts rapidly to shocks and drifts, and (iii) yields transparent, attribute-level price explanations. The results demonstrate that interpretability and efficiency in autonomous pricing agents can be achieved jointly through structured, attribute-driven representations.

KEYWORDS

Dynamic pricing; Low-dimensional demand; Additive Feature Decomposition (AFD); Dynamic Markets; Interpretability; Attribute-level prices; LEARN

ACM Reference Format:
Srividhya Sethuraman and Chandrashekar Lakshminarayanan. 2026. Learning to Price: Interpretable Attribute-Level Models for Dynamic Markets. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/FJUA8337

1 INTRODUCTION

E-commerce platforms operate in complex, high-dimensional markets where product variety is vast and consumer behavior is dynamic. They rely on sophisticated tools for recommendation [18, 21], inventory-aware personalization [15], and demand forecasting [24]. Among these tools, pricing remains a central and powerful lever, affecting not only revenue but also inventory movement, competition, and customer engagement. Dynamic pricing, the practice of adjusting prices over time in response to market conditions, enables platforms to align prices with consumer preferences, temporal trends, and competitive pressures [6, 10].

The dynamic pricing problem involves sequentially setting prices for products with the goal of maximizing cumulative revenue over time under uncertain and evolving demand. Dynamic pricing strategies can broadly be categorized into two paradigms. The first, demand-based pricing, adjusts product prices in response to aggregate demand signals, offering a coarse-grained control mechanism over revenue and inventory [7]. In contrast, personalized dynamic pricing leverages fine-grained user-level features — such as demographics, purchase history, and behavioral patterns — to set individualized prices aimed at maximizing user-specific revenue or engagement [5, 16]. While demand-based methods are reactive and population-level, personalized approaches enable more nuanced, context-aware pricing decisions.

In this paper, we focus on demand-based pricing. The demand for a product is typically decomposed into (i) a *baseline* component—representing consumer preference over product attributes, and (ii) an *elasticity* component that quantifies how demand changes in response to pricing [4, 14]. Self-price elasticity captures the effect of a product's price on its own demand, while cross-price elasticity reflects the influence of other products' prices, particularly for substitutes with similar features [2]. A key aspect in modeling elasticity is the substitution effect, wherein a similar product priced lower adversely affects the demand of a given product.

Dynamic pricing in real-world markets is challenging due to two key factors: (a) the *high dimensionality* arising from large product catalogs and (b) the *uncertainty* in baseline preferences and elasticity matrices. An autonomous pricing agent that uses (a) a low-dimensional feature representation to tackle the high dimensionality of product catalogs and (b) an online learning paradigm to tackle the uncertainty in elasticity is a good candidate to solve the dynamic pricing problem. A highly desirable quality of such an autonomous agent is interpretability: its decisions should provide actionable business insights.

In this paper, we study dynamic pricing in markets with closely related products — such as mobile phones, hotels, or airline tickets — where both demand and pricing are driven by product attributes. In this setting, we make two key contributions: (I) a novel interpretable low-dimensional demand model, and (II) a dynamic pricing algorithm tailored for this model under unknown and evolving market conditions.

---

[*] Srividhya Sethuraman is a Part-time student in IIT Madras and a Researcher at TCS Research, India.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/FJUA8337

Contribution I: Our notion of interpretability is the ability to explain price differences through observable attribute contributions and substitution effects. Informally, we aim to answer the following question:

Why is product A priced higher than product B?

We aim to develop an interpretable model that can answer:

"Product A is priced higher because it offers additional features that product B lacks."

To support such reasoning, we propose the Additive Feature Decomposition based Low-Dimensional (AFDLD) model, which expresses prices as additive contributions from observable attributes. By linking prices directly to measurable features (e.g., RAM, camera, display), AFDLD provides a transparent decomposition of each product's price and captures substitution effects among similar items. This constitutes our first contribution: an interpretable framework that explains price differences and substitution patterns through attribute-level decomposition.

Contribution II: Next, under the AFDLD model, we address the question:

How should we price products in dynamic markets?

Real-world markets are dynamic, with shifting demand and uncertain consumer preferences. To tackle this, we introduce ADEPT—an online learning algorithm for dynamic pricing under the AFDLD model. ADEPT operates in attribute space and achieves sublinear regret of $\tilde{O}(\sqrt{d}T^{3/4})$, enabling effective pricing decisions in evolving environments.

Novelty: Mueller et al. [19], introduce a low-rank demand model and OPOK, an online learning algorithm for dynamic pricing. In comparison, our work is novel in two respects - (a) Our Additive Feature Decomposition (AFDLD) enables attribute-level interpretability and (b) we explicitly incorporate substitution effects into the demand formulation. To the best of our knowledge, existing low-rank bandit formulations do not explicitly model substitution effects across products in an interpretable manner. This omission is critical, as substitution drives cross-product interactions that significantly influence demand and revenue outcomes. Our work addresses this gap by incorporating substitution effects directly into the attribute-level pricing formulation, thereby enabling realistic and actionable pricing.

We next introduce Additive Feature Decomposition (AFD) to model interpretable, attribute-based prices.

2 ADDITIVE FEATURE DECOMPOSITION (AFD) OF PRODUCT PRICES

We begin with the definition of attribute-based features.

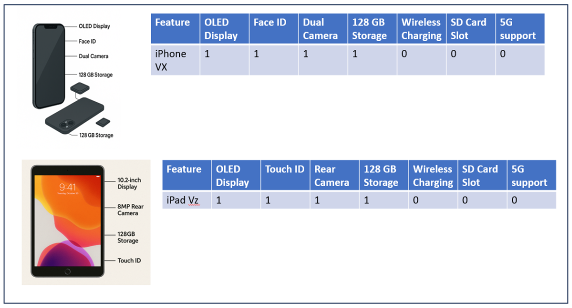
Definition 1 (Product–Feature Matrix). *For a market with N products, each having d observable attributes, let $U \in \mathbb{R}^{N\times d}$ denote the product–feature matrix. The feature vector of the i^{th} product is $U(i) = (U(i,1),\ldots,U(i,d))$, where for $j = 1,\ldots,d$, $U(i,j) \in \{0,\ldots,k\}$, i.e., each attribute takes a discrete value from $\{0,\ldots,k\}$. We also use $u(i)^\top \in \mathbb{R}^d$ to denote the i^{th} row of $U \in \mathbb{R}^{N\times d}$.*

Additive feature decomposition (AFD). AFD expresses the price of a product as the *sum of interpretable feature contributions*: each product/service attribute (size, brand, materials, options, policies) carries a transparent attribute-level price. We now discuss how Additive Feature Decomposition (AFD) naturally arises in four domains: electronics, apparel, hotels, and airlines. (See Figure 1 [1]).

Figure 1: Illustration of additive feature-based price decomposition across domains.

Examples Include:

1. **Consumer electronics:** phone/tablet prices decompose into base hardware, storage tier, camera/display modules, and software/services premia (Figure 2 [2]).
2. **Apparel:** garments combine base fabric, workmanship, certification (e.g., organic), brand price, and seasonal exclusivity.
3. **Hotels:** a nightly rate splits into base room, location/view, board plan, amenities, and flexibility options.
4. **Airlines:** fares add base transport to taxes/fees, baggage, seat selection, lounge/service upgrades.



Feature	OLED Display	Face ID	Dual Camera	128 GB Storage	Wireless Charging	SD Card Slot	5G support
iPhone VX	1	1	1	1	0	0	0

Feature	OLED Display	Touch ID	Rear Camera	128 GB Storage	Wireless Charging	SD Card Slot	5G support
iPad Vz	1	1	1	1	0	0	0

Figure 2: Illustration of AFD representation where interpretable attributes combine additively to form transparent, feature-level price components.

Formal model. Let $u(i)^\top \in \mathbb{R}^d$ be the i^{th} row of the feature matrix $U \in \mathbb{R}^{N\times d}$ denoting the features of the i^{th} product. Let the attribute prices be denoted by $\theta \in \mathbb{R}^d$. Under AFD, the price $p(i)$ of a product i is given by,

$$p(i) = \langle u(i), \theta \rangle = u(i)^\top \theta = \sum_{j=1}^{d} u(i,j)\theta(j),$$

---

[1] All figures in Figure 1 have been taken from https://unsplash.com/.

[2] All figures in Figure 2 have been taken from https://unsplash.com/.

so each $u(i,j)\theta(j)$ is an explicit, controllable price addend. Here, each product price is directly expressed in terms of its attribute prices. This linear and interpretable structure underpins our algorithmic design: we learn (or adjust) θ online under box constraints, enabling scalable, real-time pricing.

3 RELATED WORK

Most closely related work is that of [19]. We have briefly mentioned the novelty of our work over their work in the introduction. A detailed comparison will be made in Sections 4 and 5 where we propose our model and algorithm respectively. We now compare our work with other related work in the literature. Our work sits at the intersection of (i) dynamic pricing with contextual bandits, (ii) low-rank representations for generalization, and (iii) additive/attribute-based pricing.

Dynamic pricing with contextual bandits. Contextual bandits model feature-dependent demand and have been applied to personalized and non-stationary pricing [3, 17, 20]. While these methods provide strong guarantees, they often struggle with scalability in high-dimensional feature spaces. We build on this line of work by introducing a structured, low-dimensional attribute representation that preserves sample efficiency.

Low-rank and structured feature models. Low-rank structures improve generalization in contextual bandits and recommendations [9, 23, 26, 28]. Unlike prior formulations that focus purely on compression, our model leverages a pricing-specific inductive bias: latent drivers correspond to interpretable product attributes.

Additive and attribute-based pricing. Additive decomposition enables transparency and control in practical pricing systems [1, 8, 12, 25]. Enterprise CPQ tools (e.g., Salesforce Communications Cloud) operationalize this through *Attribute-Based Pricing* (ABP) rules [22, 27]. Our approach extends ABP into an *online learning* framework, treating attribute adjustments as decision variables and learning from bandit feedback via projection-free updates.

4 AFDLD: A NOVEL INTERPRETABLE LOW-DIMENSIONAL DEMAND MODEL

In this section, we formulate the dynamic pricing problem as a regret minimization problem. We then introduce our low-dimensional demand model which is based on the additive features discussed in the previous section. We then show that under this demand model, the regret minimization problem reduces to a convex optimization problem. We then discuss how our novel model captures the substitution effects absent in the low-rank model [19]. We also illustrate the importance of capturing substitution effects via toy examples.

Regret Formulation. We study high-dimensional dynamic pricing where a seller offers N products over T time periods. At each round t, the seller chooses a price vector $p_t \in \mathrm{R}^N$, observes demand $q_t \in \mathrm{R}^N$, and earns revenue $R_t(p_t) = -\langle q_t, p_t\rangle$. The goal is to maximize cumulative revenue while keeping the *regret* small:

$$\mathrm{Regret}(T) = \mathrm{E}\left[\sum_{t=1}^{T}\left(R_t(p^*) - R_t(p_t)\right)\right], \tag{1}$$

where

$$p^* = \arg\min_{p\in\mathcal{P}} \sum_{t=1}^{T} -R_t(p),$$

is the best fixed price vector in hindsight and $\mathcal{P}$ is the set of allowable prices. Maximizing total revenue is equivalently minimizing total of negative revenue.

We now introduce our novel **Additive Feature Decomposition-based Low Dimensional (AFDLD) demand model** . As motivated in Section 2, let $U \in \mathrm{R}^{N\times d}$ denote the product feature matrix defined in terms of the observable attributes, and $\theta_t \in [\theta_{\min}, \theta_{\max}]^d$ the attribute-level prices. Product-level prices are then given by

$$p_t = U\theta_t$$

hence additive decomposition ensures that each attribute contributes a distinct increment to the final price, providing transparency and interpretability.

Demand for product i at time t is modeled as

$$\begin{aligned} q_t(i) &= u(i)^\top z_t - \alpha_{ii}\,\langle u(i), u(i)\rangle_{V_t}\, p_t(i) \\ &\quad + \sum_{j=1}^{N} \alpha_{ij}\,\langle u(i), u(j)\rangle_{V_t}\,(p_t(j) - p_t(i)) + \varepsilon_t(i). \end{aligned} \tag{2a}$$

where $z_t \in \mathrm{R}^d$ captures baseline demand in attribute space, α_{ii} encodes own-price sensitivity, and α_{ij} models substitution strength between product i and j. The $(p_t(j) - p_t(i))$ term captures realistic substitution effects: if a close substitute j becomes more expensive, then demand for i rises and vice-versa. $V_t \in \mathrm{R}^{d\times d}$ is a positive definite matrix which captures the relative importance of attributes.

Revenue as a Quadratic Form. Since the product price is related to the base price via $p_t = U\theta_t$, collecting terms yields a quadratic revenue function in θ_t:

$$R_t(\theta_t) \;=\; -\,\theta_t^\top U^\top M U \theta_t \;+\; \theta_t^\top U^\top z_t, \tag{3}$$

where $M_t = M_1 + M_2 + M_3$ is the attribute-elasticity operator with

$$\begin{aligned} M_1 &= 2\,\mathrm{Diag}(UV_tU^\top), \\ M_2 &= -\,UV_tU^\top, \\ M_3 &= \mathrm{Diag}\left(\sum_j \langle u(i), u(j)\rangle_{V_t}\right). \end{aligned} \tag{4}$$

Proposition 1 (Convex Optimization). *The objective $-R_t(\theta_t)$ is convex in θ_t.*

Remarks:

- For fixed $z_t = z, \forall t \geq 0$ and $V_t = V, \forall t \geq 0$ the market is static and $R_t(\theta) = R(\theta), \forall t \geq 0$. Since -$R(\theta)$ is convex, in the case of static market, we can compute the optimal attribute-level price by minimizing as $\theta^* = \arg\min_\theta R(\theta)$.
- In the case when the market is dynamic, minimizing the regret in (1) amounts to an *online convex optimization problem*, which is achieved by our ADEPT algorithm (Section 5). *(See Supplement*[*3] *for an extended discussion.)*

---

[3] Link for supplement: https://arxiv.org/abs/2602.00188

Comparison with prior model. In the low-rank demand model of [19] the demand for product i is given by:

$$q_t(i) = u_t^{\text{orth}}(i)^\top z_t - \sum_{j=1}^{N} \langle u^{\text{orth}}(i), u^{\text{orth}}(j) \rangle_{V_t} \, p_t(j) + \varepsilon_t(i) \quad (5)$$

We now list the differences between our model in (2a) and that of [19] in (5), the key one being that (5) does not capture substitution effects.

- **Features:** In our model U is based on observable product features, whereas in (5), U^{orth} is an orthogonal latent feature matrix.
- **Baseline Demand:** In both models $z_t \in \mathbb{R}^d$ denotes the baseline demand. The difference is that in our model, z_t captures the baseline demand for the observable product attributes, whereas in (5), it stands for the baseline demand for the latent features.
- **Elasticity:** As discussed in the text below (2a), our model captures the substitution effects. As can be seen from (5), when two different products i, j are similar, it is quite natural to expect that if j is priced higher than i, customers will prefer i more than j which will lead to an increase in demand for i. However, in (5), an increase in price of a similar product j decreases the demand for product i which contradicts standard economic intuition.
- **Low Dimension vs. Low Rank:** The final revenue expression in our setting is low-dimensional (see (2a)) and is not low-rank. In (5), the model is low-rank, however, the price $p_t \in \mathbb{R}^N$ is not.
- **Pricing:** In this paper, we assume that the pricing is low-dimensional in that $p_t = U\theta_t$. In (5), the pricing p_t is still in N dimensions which is addressed via a projection step in their algorithm (we discuss this in the next section).

4.1 Interpretability of AFDLD Model

In markets with similar products, substitution effects strongly influence both demand and pricing. Our notion of interpretability is therefore tied to these effects—understanding how prices and demand vary with attribute-level similarity among products. The AFDLD model captures this relationship explicitly. We now illustrate how it produces interpretable and transparent pricing behavior through simple synthetic examples that reveal its attribute-level reasoning.

Setting 1: Equal attribute importance in V. To illustrate interpretability under the AFDLD formulation, we consider a realistic electronics example involving three smartphone variants (N=3) and three observable attributes (d=3): storage tier (A_1), camera module (A_2), and display type (A_3). The product–attribute matrix $U \in \mathbb{R}^{3\times3}$ and baseline demand $z \in \mathbb{R}^3$ are explicitly specified, while the feature–interaction matrix V=I isolates the role of self-effects. The elasticity coefficients are fixed at $\alpha_{ii} = \alpha_{ij} = 0.15$.

Experiment 1: Non-overlapping attributes (Table 1). In this experiment, we consider a degenerate setting in which each product has only one attribute. One can think of product P_1 being storage of 128 GB (A_1 alone), product P_2 being a dual camera (A_2 alone) and product P_3 being an OLED display (A_3 alone). In this case, for $i = 1, 2, 3$ the features $U(i)$ of product $P(i)$ are given by:

$$U(1) = [1\ 0\ 0], \quad U(2) = [0\ 1\ 0], \quad U(3) = [0\ 0\ 1].$$

When baseline demand is identical, the learned θ^* values are identical across attributes, leading to equal product prices (p=$U\theta^*$). When baseline demand is non-identical, the estimated θ^* yields distinct prices in direct proportion to the attribute-specific demand. The model reveals that the camera feature contributes most to willingness-to-pay, followed by storage and display — as reflected in the magnitude of θ^*. Such one-to-one mapping between attribute value and price component enables clear managerial explanation: product 2 is costlier precisely because consumers value its camera feature more highly.

Table 1: AFDLD model with $V = I$ and non-overlapping features.

Baseline demand z	θ^* **values**	**Final prices** $p = U\theta^*$
[60, 60, 60]	[13.72, 13.70, 13.72]	[13.72, 13.70, 13.72]
[100, 150, 250]	[22.96, 33.60, 56.96]	[22.96, 33.60, 56.96]

Experiment 2: Overlapping attributes (Table 2). We now allow shared features across products,

$$U(1) = [1\ 1\ 0], \quad U(2) = [0\ 1\ 1], \quad U(3) = [1\ 0\ 1].$$

corresponding to P1 (Storage + Camera), P2 (Camera + Display), and P3 (Storage + Display). For identical baseline demand, we obtain nearly uniform θ^* values and thus similar final prices. Under heterogeneous demand, the optimal θ^* and resulting p exhibit interpretable differentiation: the combination of high-value attributes (e.g., camera plus display) produces higher composite prices. Every price difference can be decomposed and explained through the additive structure $p_i = \langle U(i,\cdot), \theta^* \rangle$, exposing how much each feature contributes to each product's final price.

Table 2: AFDLD model with $V = I$ and overlapping features.

Baseline demand z	θ^* **values**	**Final prices** $p = U\theta^*$
[60, 60, 60]	[6.77, 6.84, 6.79]	[13.61, 13.63, 13.56]
[100, 150, 250]	[14.69, 17.78, 24.30]	[32.47, 42.08, 38.98]

Setting 2: Heterogeneous attribute importance in V. We now extend the smartphone example to study the effect of unequal attribute importance, captured by a diagonal interaction matrix $V = \text{diag}([1.5,\ 1.2,\ 1.0])$. The attributes—storage tier (A_1), camera module (A_2), and display type (A_3)—now differ in relative market influence: camera upgrades are perceived as most critical, followed by storage and display. The elasticity coefficients are fixed at $\alpha_{ii} = \alpha_{ij} = 0.15$, and the U structures remain the same as in Setting 1 to isolate the role of V.

Experiment 1: Non-overlapping attributes (Table 3). Here, we use the U matrix as in the previous setting with non-overlapping attributes. We observe that when the baseline demand is identical, the estimated θ^* produces proportional prices p. Here, even though demand is uniform, prices differ because the attributes vary in intrinsic importance encoded in V. This yields an interpretable ranking: OLED display (A_3) has the highest contribution, followed by camera and storage. When the baseline demand is non-identical, the interaction of demand and attribute significance amplifies price separation. Retailers can directly interpret these coefficients as attribute-level prices: a higher baseline demand for the "display-intensive" product magnifies the already high display value.

Table 3: AFDLD model with $V = \mathrm{diag}([1.5,\ 1.2,\ 1.0])$ with non-overlapping features.

Baseline demand z	θ^* **values**	**Final prices** $p = U\theta^*$
$[60, 60, 60]$	[8.94, 11.40, 13.50]	[8.94, 11.40, 13.50]
$[100, 150, 250]$	[15.20, 27.92, 56.78]	[15.19, 27.93, 56.78]

Experiment 2: Overlapping attributes (Table 4). To capture more realistic smartphones sharing multiple features, we use U matrix with overlapping attributes as in the previous setting corresponding to P1 (Storage + Camera), P2 (Camera + Display), and P3 (Storage + Display). When baseline demand is identical, we obtain θ^* and prices p. Even under identical demand, attribute heterogeneity causes asymmetric prices, highlighting how AFDLD decomposes product value into interpretable feature-wise premiums. Under non-identical demand, the estimated θ^* and p show that products combining more valued attributes (camera + display) consistently achieve higher prices.

Table 4: AFDLD model with $V = \mathrm{diag}([1.5,\ 1.2,\ 1.0])$ with overlapping features.

Baseline demand z	θ^* **values**	**Final prices** $p = U\theta^*$
$[60, 60, 60]$	[4.78, 5.71, 6.11]	[16.98, 32.81, 15.85]
$[100, 150, 250]$	[9.79, 15.40, 21.40]	[25.19, 36.80, 31.19]

Interpretability: Across both settings, the AFDLD model provides a clear, human–readable decomposition of prices into attribute–level components. The learned vector θ^* quantifies the monetary contribution of each observable feature (e.g., storage, camera, display), while the feature matrix U specifies how these features combine to form each product.

When $V = I$ (Setting 1), all attributes contribute equally and price differences arise solely from variations in baseline demand z, directly revealing which features are most valued by consumers. When V is heterogeneous (Setting 2), diagonal weights encode the relative market importance of each attribute, allowing ADEPT to separate the effects of baseline demand from intrinsic feature significance. Thus, the decomposition $p_i = \langle U(i, \cdot), \theta^* \rangle$ offers an interpretable pricing logic — every price differential can be traced back to specific attributes and their learned valuations—unlike low–rank latent models where such attribution is opaque.

5 ADEPT: A NOVEL DYNAMIC PRICING ALGORITHM

In this section, we present the ADEPT algorithm, which performs dynamic pricing directly in the attribute space. We describe how ADEPT initializes attribute-level parameters, perturbs them through one-point bandit feedback, and updates prices via simple clipping within feasible bounds. We highlight key differences between ADEPT and OPOK in initialization, gradient estimation, projection, and interpretability. Finally, we establish the theoretical regret bound for ADEPT, showing that it achieves a sublinear expected regret of $\tilde{O}(\sqrt{d}\,T^{3/4})$ under standard smoothness and noise assumptions, thereby confirming both efficiency and scalability.

Algorithm 1 ADEPT

```
Require: Feature matrix $U \in \mathbb{R}^{N\times d}$, baseline $\theta_{\text{base}}$, box $[\theta_{\min}, \theta_{\max}]$, step size $\eta > 0$, perturbation $\epsilon > 0$, horizon $T$
Initialize: $\theta_1 \leftarrow 0$; set lower and upper limits $\ell \leftarrow \theta_{\min} - \theta_{\text{base}}$, $u \leftarrow \theta_{\max} - \theta_{\text{base}}$
for $t = 1$ to $T$ do
    Sample a random direction $\xi_t \sim \mathrm{Unif}(\mathbb{S}^{d-1})$
    Perturb parameters: $\tilde{\theta}_t \leftarrow \theta_t + \epsilon \xi_t$
    Construct prices: $p_t \leftarrow U(\theta_{\text{base}} + \tilde{\theta}_t)$
    Observe revenue $y_t = R_t(p_t)$
    Compute one-point gradient estimate:
        $g_t \leftarrow -\frac{d\, y_t}{\epsilon}\, \xi_t$
    Update and clip within box:
        $\theta_{t+1} \leftarrow \mathrm{clip}_{[\ell,u]}(\theta_t - \eta g_t)$
end for
Output: Final prices $p = U(\theta_{\text{base}} + \theta_{T+1})$
```

Algorithm 2 OPOK (Online Pricing Optimization with Known Features)

```
Require: Step sizes $\eta, \delta, \alpha > 0$, feature matrix $U \in \mathbb{R}^{N\times d}$, initial price $p_0 \in S$
Ensure: Prices $p_1, \ldots, p_T$ to maximize revenue
Set prices $p_0 \in S$, observe $q_0(p_0), R_0(p_0)$
Define $x_1 \leftarrow U^\top p_0$
for $t = 1, \ldots, T$ do
    Draw $\xi_t \sim \mathrm{Unif}\{x \in \mathbb{R}^d : \|x\|_2 = 1\}$
    $\tilde{x}_t \leftarrow x_t + \delta \xi_t$
    Set prices: $p_t \leftarrow \textsc{FindPrice}(\tilde{x}_t, U, S, p_{t-1})$, observe $q_t(p_t), R_t(p_t)$
    $x_{t+1} \leftarrow \textsc{Projection}(x_t - \eta R_t(p_t)\xi_t, \alpha, U, S)$
end for
```

5.1 OPOK Vs. ADEPT

- **Initialization:** OPOK initializes with a feasible price vector $p_0 \in S$ and maps it into latent space $x_1 = U^\top p_0$. ADEPT initializes directly in attribute space ($\hat{\theta}_1 = 0$) with box constraints $[\theta_{\min}, \theta_{\max}]$ around a baseline θ_{base}.
- **Perturbation direction:** OPOK samples $\xi_t \in \mathbb{R}^d$ and perturbs the latent representation x_t. ADEPT samples $\xi_t \in \mathbb{R}^d$ and perturbs the attribute-level parameters θ_t.
- **Price construction:** OPOK uses $\textsc{FindPrice}(\tilde{x}_t, U, S, p_{t-1})$ to map the perturbed latent vector $\tilde{x}_t$ to a feasible price vector p_t, ensuring stability relative to past prices. ADEPT directly constructs candidate prices as $p_t = U(\theta_{\text{base}} + \tilde{\theta}_t)$, bypassing a separate feasibility solver.
- **Revenue observation:** Both algorithms post prices and observe demand/revenue.
- **Gradient estimation:** OPOK updates x_{t+1} via estimator: $x_{t+1} = \textsc{Projection}(x_t - \eta R_t(p_t)\xi_t, \alpha, U, S)$. ADEPT updates via estimator: $g_t = -\frac{d\, y_t}{\varepsilon}\, \xi_t$.
- **Projection/Constraints:** OPOK projects the latent point back into the feasible span using $\textsc{Projection}(\cdot)$. ADEPT clips attribute prices into $[\theta_{\min}, \theta_{\max}]$, enforcing economically meaningful nonnegativity and boundedness.
- **Output:** OPOK's iterates p_t are feasible prices at each step. ADEPT outputs the final feasible price vector $p = U(\theta_{\text{base}} + \hat{\theta}_{T+1})$.

Summary. OPOK and ADEPT differ in optimization domain and objective formulation — latent vs. attribute space—rendering their regret bounds and learned parameters not directly comparable.

Theorem 1. *[Regret of ADEPT] Let $U \in \mathbb{R}^{N\times d}$ with $\|U\|_{\text{op}} \le B_U$, and let the attribute box be $C \subset \mathbb{R}^d$ with radius $r_\Theta := \max_{\theta\in C} \|\theta\|_2$. Assume for all t: $A_t = U^\top V_t U \succeq 0$ with $\|V_t\|_{\text{op}} \le B_V$, $\|z_t\|_2 \le B_z$, and bandit noise that is σ^2–sub-Gaussian. With $\eta \asymp T^{-1/2}$ and $\delta \asymp T^{-1/4}$,*

$$\mathbb{E}\Big[\, \textstyle\sum_{t=1}^{T} \big(R_t(\theta^\star) - R_t(\hat{\theta}_t)\big) \Big] \;\le\; K\,\sqrt{d}\,T^{3/4},$$

where $\theta^\star \in \arg\max_{\theta\in C} \sum_{t=1}^{T} R_t(\theta)$ and

$$K \;:=\; c_0\big(B_U B_z r_\Theta \;+\; B_U^2 B_V r_\Theta^2 \;+\; \sigma\big),$$

for a universal constant $c_0 > 0$ (log factors absorbed into c_0). See Supplement[4] for the complete proof.*

6 EXPERIMENTS AND INSIGHTS

6.1 Empirical Validation of Additive Feature Decomposition in Real-World Datasets

We validate the additive feature decomposition (AFD) assumption, which is central to ADEPT's formulation, using two large-scale, real-world retail datasets namely Dunnhumby Complete Journey and H&M Personalized Fashion Recommendation. Both datasets are of the form $(x_i, y_i)_{i=1}^{n}$, where $x_i \in \mathbb{R}^d$ is the attribute based product feature (as discussed in Section 2) and y_i is the price. We hasten to mention that our goal in this section is to verify that the attributes based features are indeed useful as features, and not to validate our demand model. In order to achieve our goal, we show that the attribute based features have good power in predicting the prices via linear regression fit.

Dunnhumby Complete Journey (Figure 3): In this dataset, the d attributes are:

- `COMMODITY_DESC` (94 values),
- `SUB_COMMODITY_DESC` (736 values),
- `MANUFACTURER` (1527 values).

Therefore, x_i is a 2,357-dimensional vector. It consisted of 39,021 distinct products. The linear regression fit achieves a strong R^2 (0.6-0.8) and the learned coefficients exhibit intuitive semantic structure — e.g., premium brands and specialized commodity categories contribute positively to price, while generic or high-volume subcategories contribute negatively. Importantly, this decomposition generalizes well to unseen data, indicating that unit prices in grocery retail indeed follow an additive structure over product metadata. This is further supported by interpretable SHAP analyses and bar plots of regression coefficients, which confirm the additive contribution of each feature group to the final price.

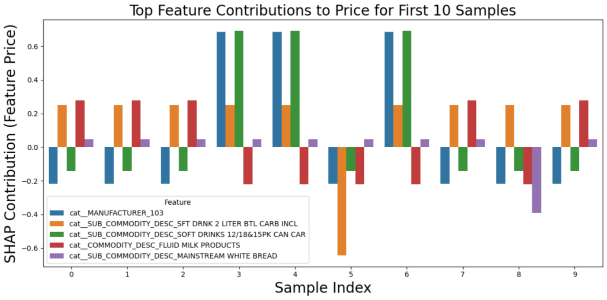


Figure 3: Additive feature contributions to unit prices in the Dunnhumby grocery dataset, obtained via linear regression coefficients.

H&M Personalized Fashion Recommendation (Figure 4): In this dataset, the d attributes are product_type_name, garment_group_name, index_name, and section which take values in one among 131 distinct types, 21 groups, 10 unique indices and 52 sections respectively. Therefore x_i is 214-dimensional vector. It consisted of 104547 distinct products. The linear regression fit reveals statistically significant coefficients across many features.

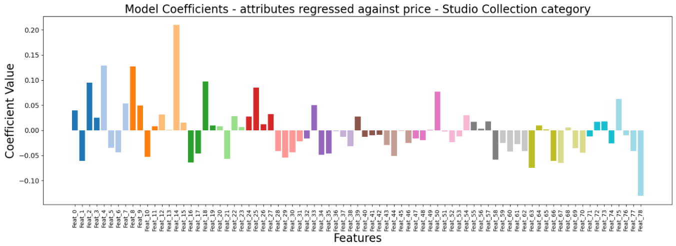


Figure 4: Additive feature contributions to prices in the H&M fashion dataset, highlighting interpretable price drivers such as garment type and season.

---

[4] Link for supplement: https://arxiv.org/abs/2602.00188

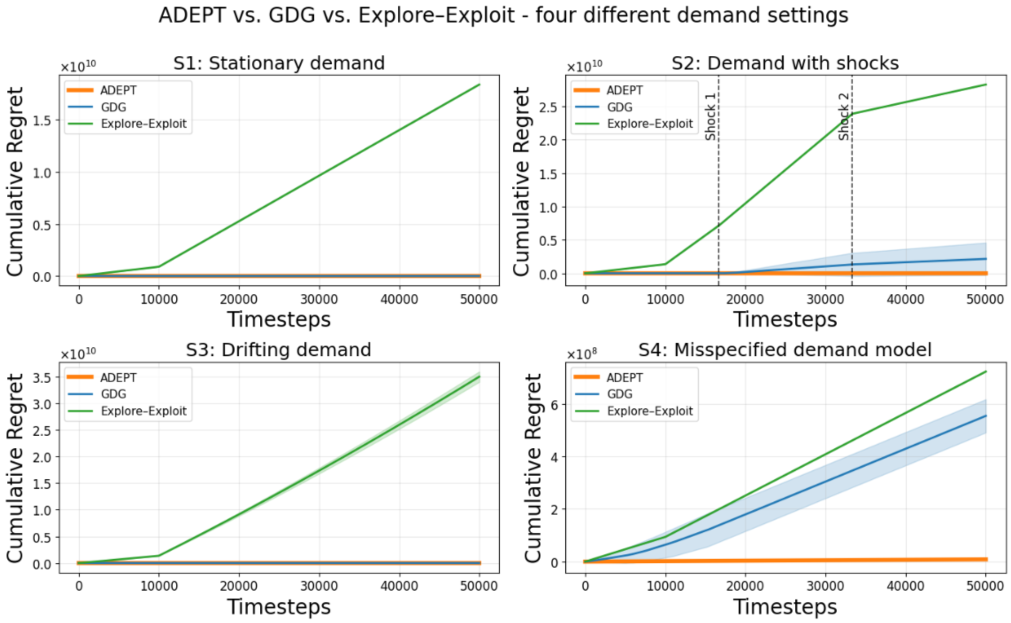


Figure 5: Cumulative regret (mean ± s.d.) over T=50,000 rounds. Top-left: S1 Stationary; top-right: S2 Shocks (vertical dashed lines mark change points); bottom-left: S3 Drifting; bottom-right: S4 Misspecified. All methods share the same price ball and comparator.

6.2 Comparing ADEPT with Baselines

Having verified additive structure in real data, we next benchmark ADEPT against existing bandit algorithms. In this section, we evaluate algorithms under the AFDLD model introduced in Section 4. We run the two-point gradient estimator of ADEPT as specified in Algorithm 1. We compare ADEPT with two benchmark algorithms namely, GDG [11] and Explore-Exploit [13]. Specifically, we compare these algorithms under different market regimes.

Common Across Settings. All algorithms are tested under the following shared configuration:

- Product-Feature matrix $U \in \mathrm{R}^{N\times d}$ with $N = 60$ products and $d = 6$ attributes, respecting additive decomposition. We divide the products into $B = 6$ disjoint blocks $\{G_b\}_{b=1}^{6}$, each containing $|G_b| = 10$ products. Each block G_b is assigned a small set of active attributes $S_b \subseteq \{1, \ldots, 6\}$, ensuring that neighboring blocks share one or more attributes to introduce controlled overlap ($S_1 = \{1, 2, 3\}, S_2 = \{2, 3, 4\}, \ldots, S_6 = \{6, 1, 2\}$). For every product $i \in G_b$ and attribute $j \in S_b$, we sample $U(i, j) \sim \text{Bernoulli}(p_b)$ with $p_b = 0.5$, and set $U(i, j) = 0$ for $j \notin S_b$. We then cap the number of active attributes per product to a small range (one or two active attributes per row). If a row has no active attributes, we activate one at random within S_b; if it exceeds the cap, we randomly retain only two active attributes.
- Time horizon $T = 50{,}000$ rounds.
- Gaussian observation noise with variance $\sigma^2 = 0.5$.
- Evaluation metric: cumulative regret relative to the best fixed price in $\mathcal{B}_P(R)$. We fix $\mathcal{B}_P(R) = \{\, p \in \mathrm{R}^N : \|p - p_0\|_2 \le 5 \,\}$ for both GDG and Explore–Exploit (EE) algorithms. The baseline price vector p_0 and the attribute-level base parameter θ_0 are related through $p_0 = U\theta_0$. We set $\theta_{\min} = \theta_0 - r$ and $\theta_{\max} = \theta_0 + r$, ensuring that the attribute parameters explore the same range as the price space defined by $B_P(R)$ ($r = 5$). For ADEPT, cumulative regret is measured relative to the best attribute-level prices (θ^*).

Algorithm-Specific Parameters.

- **GDG (Gradient Descent with Bandit Feedback) :** Operates in the orthonormal span O obtained from QR decomposition $U = OP$, with one-point random-direction smoothing and projection in O-space. See Supplement for pseudocode.
- **Explore–Exploit :** Alternates between uniform exploration in $\mathcal{B}_P(R)$, ridge regression fit of a quadratic surrogate, and trust-region exploitation. See Supplement for pseudocode.
- **ADEPT:** Updates attribute prices directly in feature space via the gradient estimator with feasibility enforced by box constraints on $\hat{\theta}_t$.

Demand Regimes. We consider four demand regimes. Each regime differs in the temporal evolution of the baseline demand z_t and the cross-elasticity operator V_t in the demand model specified in Equation (2a), while all other parameters remain fixed.

(1) **Stationary demand (S1):** Both baseline demand and cross-elasticities remain fixed, i.e. $z_t = z$, $V_t = V$ for all t. This corresponds to a stable market with no temporal variation in consumer preferences or substitution effects. In **S1**, ADEPT maintains nearly flat regret, GDG grows slowly due to variance in one-point updates, and Explore-Exploit accumulates error from surrogate bias.

(2) **Structural shocks (S2):** At $t = T/3$ and $t = 2T/3$, the baseline and elasticity parameters (z_t, V_t) are generated again with the same assumptions independently in each new phase. This simulates sudden market shocks such as competitor entry or seasonal change. In **S2**, ADEPT adapts rapidly after each shock, while GDG shows pronounced regret spikes and Explore-Exploit requires many rounds to re-learn post-shock surrogates.

(3) **Drifting demand (S3):** Parameters evolve gradually according to stochastic drift:

$$z_{t+1} = z_t + w_t, \qquad V_{t+1} = \Pi_{\mathcal{V}}(V_t + W_t),$$

where $w_t \sim \mathcal{N}(0, I)$ and $W_t \sim \mathcal{N}(0, 0.1I)$, and $\Pi_{\mathcal{V}}$ projects into the positive-definite cone. This captures smooth preference shifts over time.

(4) **Misspecified demand (S4):** Instead of low-rank V_t, the true demand operator is taken to be full-rank with heterogeneous eigenvalues. This explicitly violates the low-rank assumption and tests robustness of the algorithms under model misspecification. In **S4**, ADEPT remains competitive despite misspecification, while GDG and Explore-Exploit incur higher regret due to structural mismatch with the full-rank M_t.

In all experiments, ADEPT's attribute-space updates with a structured U deliver lower cumulative regret and stronger adaptability. Finally, to verify consistency between empirical and theoretical performance, we estimate ADEPT's regret-growth rate through tail-slope analysis in the next subsection.

6.3 Empirical rate verification via tail–slope fitting

We verify that ADEPT attains the rate in Theorem 1. If $R(t) \asymp t^{\alpha}$, then $\log R(t) = a + \alpha \log t$ is linear in $\log t$ with slope α. To estimate α, we fit a *tail secant* over the last $\rho = 0.5$ of the horizon ($T = 50{,}000$):

$$\widehat{\alpha}_{\text{tail}} = \frac{\log R(T) - \log R(t_0)}{\log T - \log t_0}, \qquad t_0 = \lfloor (1-\rho)T \rfloor.$$

Results (Fig. 6). The fitted slope, $\widehat{\alpha}_{\text{tail}} = 0.742$, closely matches the theoretical $t^{3/4}$ rate. Early curvature arises from pre–asymptotic effects. For (N=60, d=6, T=5e^4), the tail is nearly linear, confirming that for ADEPT empirically achieves $\tilde{O}(T^{3/4})$ regret. [5]

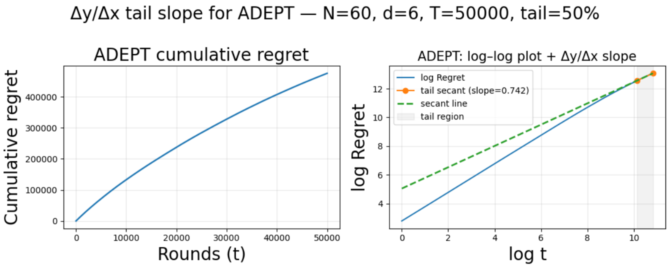


Figure 6: Tail–slope analysis for ADEPT. Left: cumulative regret $R(t)$. Right: log–log plot of $R(t)$ with the tail window (gray) and secant fit; estimated slope $\widehat{\alpha}_{\text{tail}} = 0.742 \approx 3/4$.

7 CONCLUSION

In this paper, we considered dynamic pricing in markets with similar products in which the product attributes affect the demand and pricing. For this setting, we proposed the **Additive Feature Decomposition-based Low-Dimensional Demand (AFDLD)** model, which expresses prices as additive functions of observable attributes while capturing substitution effects among related products. Building on this, **ADEPT** — a projection-and gradient-free online learner — optimizes directly in attribute space and achieves a sublinear regret of $\tilde{O}(\sqrt{d}\,T^{3/4})$. Experiments across stationary, shocked, drifting, and misspecified regimes show that ADEPT attains low regret, rapid adaptation, and interpretable attribute-level price explanations. These results establish that interpretability and efficiency can coexist in dynamic pricing.

Future work will extend AFDLD to nonlinear and personalized settings, evolving attribute spaces, and multi-agent markets, advancing toward autonomous and explainable pricing systems.

REFERENCES

[1] Bozhidar Aleksandrov, Anna Leontjeva, Tasha Hemphill, and Ulrich Paquet. 2023. Modular pricing of digital subscription bundles. *Journal of Revenue and Pricing Management* 22, 1 (2023), 33–48.
[2] Michelle Andrews and Imran S Currim. 2016. Estimating cross-price elasticities from aggregate retail data. *Journal of Marketing Research* 53, 2 (2016), 187–198.
[3] Gah-Yi Ban, Narges Keskin, and Zhaoxu Liu. 2021. Personalized pricing with nonparametric demand learning. In *Proceedings of the 38th International Conference on Machine Learning*. 517–527.
[4] Steven Berry, James Levinsohn, and Ariel Pakes. 1995. Automobile prices in market equilibrium. *Econometrica* 63, 4 (1995), 841–890.
[5] Lijie Chen and Guillermo Gallego. 2019. Personalized dynamic pricing with machine learning. *Handbook of Pricing Analytics* (2019), 1–29.
[6] Liyuan Chen, Alan Mislove, and Christo Wilson. 2016. An empirical analysis of algorithmic pricing on Amazon Marketplace. In *Proceedings of the 25th International Conference on World Wide Web*. 1339–1349.
[7] Xi Chen, Guillermo Gallego, and Anant Sinha. 2016. Dynamic pricing with demand learning and reference effects. In *Advances in Neural Information Processing Systems (NeurIPS)*.
[8] Xinyi Chen, Jun Zhang, and Jie Li. 2021. Interpretable learning for travel demand and pricing with additive models. In *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1072–1080.
[9] Alon Cohen, Elad Hazan, and Tomer Koren. 2016. Contextual bandits with latent confounders: An NMF approach. In *Proceedings of the 33rd International Conference on Machine Learning*.
[10] Wedad Elmaghraby and Pinar Keskinocak. 2003. Dynamic pricing in the presence of inventory considerations: Research overview. *Production and Operations Management* 12, 4 (2003), 535–552.
[11] Abraham D. Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. 2005. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 385–394.
[12] Ananya Gupta, Pradeep Chintagunta, and Dipak Jain. 2020. Pricing airline ancillaries: A discrete choice modeling approach. In *Proceedings of the 2020 INFORMS Annual Meeting*.
[13] Mikael Henaff. 2019. Explicit explore-exploit algorithms in continuous state spaces. *Advances in Neural Information Processing Systems* 32 (2019).
[14] Daniel Hosken, David Reiffen, Andrew Varney, and Luke Taylor. 2008. Retail price drivers and retail pricing strategies. *Journal of Economics & Management Strategy* 17, 1 (2008), 147–187.
[15] Dietmar Jannach and Gediminas Adomavicius. 2016. Recommendation systems for e-commerce. *User Modeling and Personalization* (2016).
[16] Jaya Kawale, Trapit Bansal, Sanjay Chawla, Aditya Sharma, Branislav Kveton, Georgios Theocharous, and Dennis Wilkinson. 2015. Price Optimization for Revenue Management using Reinforcement Learning. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM)*.
[17] Narges Keskin and Assaf Zeevi. 2020. Dynamic pricing with demand learning and reference effects. *Operations Research* 68, 5 (2020), 1381–1405.

---

[5] The per-epoch runtime comparison of ADEPT with OPOK, GDG and EE is given in supplement. Link for supplement: https://arxiv.org/abs/2602.00188.

[18] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[19] Jonas W Mueller, Vasilis Syrgkanis, and Matt Taddy. 2019. Low-rank bandit methods for high-dimensional dynamic pricing. *Advances in Neural Information Processing Systems* 32 (2019).

[20] Yutian Qi, Weinan Dong, Kaixiang Shen, Zhi Tang, and Hongyuan Zha. 2021. Meta-learning for contextual bandit exploration. In *Advances in Neural Information Processing Systems*.

[21] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. *Recommender systems handbook*. Springer.

[22] Salesforce. 2025. Attribute-Based Pricing. https://help.salesforce.com/s/articleView?id=ind.comms_attribute_based_pricing.htm&type=5. Salesforce Help, Communications Cloud. Accessed: 2025-10-01.

[23] Rahul Sen, Bipul Mathew, Yash Bhalgat, Prateek Jain, and Manik Varma. 2016. Contextual bandits with latent confounders: Model and algorithm. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*.

[24] Matt Taddy, Matt Gardner, Liyuan Chen, and David Draper. 2019. A nonparametric Bayesian analysis of heterogeneous treatment effects in digital marketing. *Journal of Marketing Research* (2019).

[25] Meng Tang, Yile Lu, Xiaoyi Huang, Meng Sun, and Yu Wang. 2022. Personalized and interpretable pricing for fashion e-commerce using multimodal decomposition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

[26] Zhen Wang, Haoran Liu, Lihong Li, and Qiang Liu. 2021. Low-rank bandits with side observations. In *Proceedings of the 38th International Conference on Machine Learning*. 10766–10776.

[27] Mengzhenyu Zhang, Christopher Ryan, and Wei Sun. 2022. Attribute-based Pricing: A nested-logit approach. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.4258247

[28] Xiaohan Zhu, Yifan Huang, Bo Lin, and Yichong Wu. 2021. Improving demand forecasting in e-commerce with attention-based low-rank embeddings. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3780–3790.