Nonparametric Pricing Bandits Leveraging Informational Externalities to Learn the Demand Curve

Ian N. Weaver
National University of Singapore Business School; i.weaver@nus.edu.sg

Vineet Kumar
Yale School of Management; vineet.kumar@yale.edu

Lalit Jain
Foster School of Business, University of Washington; lalitj@uw.edu

We propose a novel, theory-based approach to the reinforcement learning problem of maximizing profits when faced with an unknown demand curve. Our method, rooted in multi-armed bandits, balances exploration and exploitation across various prices (arms) to maximize rewards. Traditional Gaussian process bandits capture one informational externality in price experimentation – correlation of rewards through an underlying demand curve. We extend this framework by incorporating a second externality, monotonicity, into Gaussian process bandits by introducing monotonic versions of both the GP-UCB and GP-TS algorithms. Through reduction of the demand space, this informational externality limits exploration and experimentation, outperforming benchmarks by enhancing profitability. Moreover, our approach can also complement methods such as partial identification. Additionally, we present algorithm variants that account for heteroscedastic noise in purchase data. We provide theoretical guarantees for our algorithm, and empirically demonstrate its improved performance across a broad range of willingness-to-pay distributions (including discontinuous, time-varying, and real-world) and price sets. Notably, our algorithm increased profits, especially for distributions where the optimal price lies near the lower end of the considered price set. Across simulation settings, our algorithm consistently achieved over 95% of the optimal profits.

Key words: Multi-armed Bandits, Reinforcement Learning, Gaussian Processes, Pricing

1. Introduction

We propose a new method based on reinforcement learning using multi-armed bandits (MABs) for pricing by efficiently learning an unknown demand curve. Our algorithm is a nonparametric method which incorporates microeconomic theory into Gaussian process Thompson sampling. Specifically, we impose weak restrictions on the monotonicity of demand, creating informational externalities between the outcomes of arms in the MAB. Our method exploits these informational externalities to achieve more efficient experimentation, higher profitability, and enables more flexible price sets than current methods.

Consider a manager tasked with determining the price for a product or service. A natural starting point is to use knowledge of the demand curve. However, the demand curve is typically known only within the limited price range where an established product has been sold, leaving it unknown elsewhere. Consequently, pricing errors are most pronounced for new products that differ significantly from past offerings (Huang et al. 2022). Even in well-established categories, demand can change substantially across markets and time. A global survey of sales VPs, CMOs, and CEOs at more than 1,700 companies conducted by Bain & Company highlights the insidious impact of poor pricing, which can cause long-term and persistent damage to a firm's financial prospects (Kermisch and Burns 2018).

To learn the demand curve, a common approach is A/B experimentation (Furman and Simcoe 2015), which randomly allocates consumers across a set of different prices, typically in a balanced experiment. However, this approach presents several challenges. First, firms across a wide range of industries are often reluctant to conduct extensive price experimentation due to frictions (Aparicio and Simester 2022) or risks. For example, price experimentation can confuse consumers or alter their expectations leading to uncertain gains (Dholakia 2015).[1] Second, firms often do not run experiments for long enough durations, making it difficult to detect effects (Hanssens and Pauwels 2016). Third, even in well-known categories, demand can undergo significant changes over time, necessitating periodic relearning of demand.

These factors directly imply that a method capable of efficiently identifying and learning the critical part of the demand curve with minimal experimentation can be highly valuable, particularly for identifying the optimal price points for products. This problem is especially suited to a reinforcement learning method known as multi-armed bandits (MABs), which simultaneously learn while earning, thereby avoiding the wasteful explorations associated with A/B testing. MABs address the classic trade-off between *exploitation* (maximizing current payoff) and *exploration* (gathering additional information) as the agent seeks to maximize rewards over a given horizon.

Two common MAB algorithms are Upper Confidence Bound (UCB) and Thompson Sampling (TS). The UCB algorithm (Auer 2002, Auer et al. 2002) constructs an upper confidence bound for the reward of each arm by combining the estimated mean reward

[1] It is noteworthy that pricing differs from advertising, where companies are generally more willing to experiment (Simester et al. 2009, Huang et al. 2018, Sahni and Nair 2020).

with an exploration bonus. This balance between exploiting high-payoff arms and exploring less-certain ones is achieved by selecting the arm with the highest upper confidence bound. Meanwhile, the TS approach involves Bayesian updating of the reward distribution for each arm as they are played (Thompson 1933). Arms are chosen probabilistically, with those having higher means being more likely to be selected. Thus, TS is a stochastic approach, whereas UCB is deterministic.

One notable feature of UCB and TS is that they treat the rewards from arms as independent. However, in pricing, this assumption does not hold, as an underlying demand curve connects the different arms (prices). In this paper, we propose a new MAB method that builds upon canonical bandits by leveraging two distinct but related sources of *informational externalities*, informed by economic theory on demand curves.

The first informational externality we observe is that rewards are correlated across prices through the demand function. Specifically, demand at closer price points is more likely to be similar than demand at more distant points. This means that the demand at a focal price point provides insights not only about that price but also about others. This relationship has been previously recognized in bandit literature and implemented using a Gaussian process (GP) combined with baseline bandit methods (Ringbeck and Huchzermeier 2019). Modeling demand with a GP allows for the learning of a more general reward function, rather than restricting learning to rewards associated with specific arms. We empirically build on this research through a systematic study of its impact across various settings.

The second informational externality is the characterization that aggregate demand curves are monotonic, consistent with microeconomic theory. Specifically, the quantity demanded at a focal price must be *weakly lower* than the demand at all prices below the focal price. However, imposing monotonic shape restrictions in the GP space is not trivial; addressing this challenge constitutes the main contribution of our paper. We leverage the idea that sign restrictions are easier to impose. The connection between the two is established by noting that: (i) a monotonically decreasing function is equivalent to its first derivative being negative at every point, and (ii) the derivative of a GP is also a GP. Critically, this transforms the problem from imposing monotonic shape restrictions in the original GP space to imposing sign restrictions in the GP space of derivatives.

We make the following contributions. To researchers and practitioners, we provide a method that builds upon Gaussian process bandits by specifying demands curves to be downward sloping without parametric restrictions. Our algorithm efficiently obtains only monotonic, downward-sloping demand curves throughout the price experimentation process. We demonstrate the effectiveness of the algorithm through theoretical analysis, and by evaluating performance relative to benchmarks for a wide range of conditions, including from field data. Our approach results in significantly higher profits relative to state-of-the-art methods in the MAB literature. Furthermore, we characterize the underlying willingness-to-pay distributions under which our algorithm delivers greater performance improvements and explores the mechanisms driving these results. Overall, the increase in efficiency means that a larger number of prices can be included in the consideration set. These are important managerial considerations, given the frictions in undertaking pricing experimentation. More broadly, in other situations where data has some general known form a priori, our approach shows how such constraints can be combined with the flexibility of nonparametric bandits to improve empirical performance.

Our method offers several advantages relative to existing approaches for optimal pricing under unknown demand. The primary advantage is its efficiency in achieving optimal pricing by learning demand in a flexible, nonparametric manner, particularly when the firm lacks well-defined prior information. If a firm is willing to undertake adaptive experimentation to learn demand while minimizing the cost of experimentation, our method enables the firm to achieve higher profitability. A second advantage is the method's ability to systematically incorporate theoretical knowledge into reinforcement learning problems, ensuring that the resulting demand satisfies theoretical constraints. Furthermore, it can be applied to learn about consumer valuations for any vertical quality-like attribute, not just prices. Third, our method provides an estimate of uncertainty in addition to point estimates of the demand curve. Indeed, the entire posterior distribution can be obtained, offering a complete characterization of uncertainty around the learned demand curve. Importantly, this approach also provides demand estimates for prices outside the initial price set chosen. Fourth, the method requires little to no human judgment. Unlike most typical RL models, hyperparameter tuning is automatic, and the method is computationally tractable, allowing the bandit to operate in real time. Fifth, our method does not rely on any knowledge of consumer characteristics, unlike partial identification approaches such as UCB-PI (Misra et al. 2019). It can, however, be used in conjunction with partial identification methods to enhance performance.

We derive theoretical regret bounds for GP-TS when monotonicity is incorporated. Our analysis follows the standard approach of Srinivas et al. (2009) and achieves similar results. However, by enforcing monotonicity, the space of feasible demand curves is reduced, which in turn decreases the path-dependent regret term and results in tighter regret bounds.

Empirically, we run simulations across a range of settings and find that our proposed algorithm outperforms several state-of-the-art benchmarks, including UCB, TS, N-TS, GP-UCB, and GP-TS. Averaged across three main underlying willingness-to-pay distributions, our algorithm achieves the highest performance, reaching 92% of optimal profits after 500 consumers and 97% of optimal profits after 2500 consumers. Furthermore, we observe that profits consistently increase when monotonicity is incorporated into GP-UCB (by 21-26% after 500 consumers and 5-8% after 2500 consumers) and GP-TS (by 10-14% after 500 consumers and 4-6% after 2500 consumers), regardless of the price set granularity (number of arms). However, there is substantial heterogeneity in the observed uplifts, depending on the underlying distribution. The largest gains occur when the optimal price lies near the lower end of the price set. This variation in uplifts is driven by heterogeneity in benchmark performance across underlying distributions, whereas our algorithm performs consistently.

We also test additional challenging scenarios. First, we evaluate the case where demand exhibits a discontinuity motivated by the left-digit bias, which presents a challenge since a GP assumes continuity. We find that our proposed method continues to perform best. However, if managers are particularly concerned about a sufficiently large left-digit bias, they may wish to focus only on prices at the discontinuities. Next, we consider cases where long-term advantages might arise from using our method. Specifically, we examine situations where demand varies with seasonality and find that, even when these shifts are unpredictable, our proposed method achieves a long-run performance advantage over the benchmarks. Further, we demonstrate that these advantages persist in real-world settings, using a valuation distribution estimated from data on demand for a streaming service. Finally, we develop an alternative specification that accounts for heteroscedastic noise in the purchase input data, resulting in small additional performance gains.

2. Literature Review

Our research is related to several streams detailed below. The setting features pricing experimentation and learning over time, which is an active area of research across marketing, operations research, economics and computer science.

Demand Learning

Studies in marketing and economics typically make strong assumptions about the information that a firm has regarding product demand. The strongest assumption used is that the firm can make pricing decisions based on knowing the demand curve (or WTP) (Oren et al. 1982, Rao and Bass 1985, Tirole 1988). A generalization of this assumption is that firms know the demand only up to a parameter, used in some of the earlier works on learning demand through price experimentation (Aghion et al. 1991, Rothschild 1974). Typically, a consumer utility function is specified in terms of product characteristics, price, and advertising, and preference parameters are estimated from data (Zhang and Chung 2020, Jindal et al. 2020, Huang et al. 2022). However, all these models predetermine the shape of the demand curve, and cannot incorporate all possible demand curves. Nonparametric approaches are the gold standard, and have been used to account for state changes between periods, but are often simplified substantially (e.g. having two periods) to ensure analytical tractability (Bergemann and Schlag 2008, Handel and Misra 2015). However, this stream typically does not consider learning through active experimentation, which is the focus of multi-armed bandits stream below.

Multi-armed Bandits

Multi-armed bandit (MAB) methods are an active learning approach based on reinforcement learning and are used across many fields, with business applications in advertising (Schwartz et al. 2017), website optimization (Hill et al. 2017, Hauser et al. 2009), and recommendation systems (Kawale et al. 2015). Two fundamental arm selection algorithms (or decision rules) that form the foundation for MAB methods are (a) Upper Confidence Bounds (UCB), based on Auer et al. (2002), which is a deterministic and (b) Thompson sampling (TS), based on Thompson (1933), a stochastic approach.

Several previous works have explored dynamic pricing with non-parametric demand assumptions. These often enforce conditions like smoothness or unimodality, and then approximate the demand using locally parametric functions, such as polynomials (Wang et al. 2021). However, the resulting algorithms depend heavily on these specific choices. In contrast, our approach imposes no constraints beyond the economically motivated monotonic shape constraint. Any further restrictions arise naturally from the chosen kernel (e.g., RBF or Matern), offering greater flexibility for practitioners hesitant to make strong assumptions.

While shape constraints have been considered in other contexts, their application to pricing and reward maximization remains limited. Thresholding bandits aim to find the arm closest to a given threshold, assuming monotonic arm rewards, but do not focus on maximizing rewards (Cheshire et al. 2020). Other works primarily address the estimation problem (Guntuboyina and Sen 2018), or consider contextual settings with monotonic mean distributions across arms (Chatterjee and Sen 2021). These approaches, while valuable, do not directly translate to our pricing problem with a focus on reward maximization under shape constraints motivated by economic theory.[2]

However, traditional bandit methods typically only model rewards for individual arms, but not any dependencies across arms. In pricing applications, this independence ignores the information from the underlying demand curve, which we term as informational externalities. Recent research below has tried to develop methods to address this issue.

Gaussian Processes and Bandits: Gaussian Processes (GPs) are well-regarded as a highly flexible nonparametric method for modeling unknown functions (Srinivas et al. 2009). GPs provide a principled approach to allowing dependencies across arms, without restricting the functional form. Multi-armed bandits combining GPs along with a decision rule like UCB or Thompson sampling (TS) have been proposed (Chowdhury and Gopalan 2017).

More specifically, a closely related paper by Ringbeck and Huchzermeier (2019) uses GP-TS for a multi-product pricing problem. The GP is modeled here at a demand level, which is important for two reasons. First, it allows modeling demand-level inventory constraints in a multi-product setting, and second, it allows for a separation of the learning problem (at demand level) from the rewards optimization (reward is the product of demand and price). Leveraging the first informational externality, they find that GP-TS improves performance over TS. However, much is not known about the conditions (e.g. the number of arms or WTP distribution or stability of the demand curve) under which the informational externality modeled by GP-TS is important, with a material impact on profit outcomes. Our focus here is on incorporating monotonicity of demand curves, a theory-based restriction that forms the second informational externality, into GP-TS and GP-UCB models, and evaluating the resulting models. We also evaluate the value of these informational externalities across a wide spectrum of conditions.

[2] A separate line of work incorporates inventory constraints into pricing models Chen et al. (2019), Miao and Wang (2024). These studies often introduce constraints on revenue or utilize exploration phases that involve significant price fluctuations, which may not be feasible in practice. Further, their solutions are typically driven by the characterization of the optimal posted price, differing from our approach.

There are other approaches to ruling out dominated prices (arms) in pricing bandits. One particularly notable contribution that incorporates knowledge into bandits is the partial identification method (Misra et al. 2019). While this approach leverages weakly decreasing demand curves, it does not otherwise rely on dependencies across arms. Partial identification formalizes the idea that the rewards from a specific arm (price) can be dominated by those of another arm and critically depends on the availability of highly informative segmentation data to estimate demand bounds for each segment. These demand bounds are then aggregated across all segments to derive the corresponding aggregated reward bounds for each price arm. Dominated prices are eliminated if the upper bound for one price is lower than the lower bound for another price. In contrast, our algorithm does not require segmentation data to exploit the gains from the monotonicity assumption. Since the mechanisms in these approaches differ, partial identification can complement our method, enabling the creation of a hybrid approach.

3. Pricing Problem and Simple Example

3.1. Pricing Problem

We address the pricing problem faced by a firm aiming to maximize cumulative profits while experimenting under an unknown demand curve. Potential buyers (consumers) arrive sequentially and are presented with a price selected by the firm. Each consumer then decides whether to purchase a single unit or not (in line with Misra et al. (2019), we focus on discrete choice purchases), depending on whether the offered price is below or above their willingness to pay (WTP).[3] For each consumer, the reward (profit) received by the firm equals the price minus the cost if the consumer makes a purchase and zero otherwise.

Overall, the firm must make two key decisions. First is the selection of the set of prices to be tested in the experiment—this is assumed to be exogenous, although in our simulations we test multiple price sets with varying granularities to guide firms in this decision. Second, the firm is allowed to periodically adjust the price shown to an incoming consumer based on limited past purchase decisions obtained during the pricing experiment. This paper focuses on designing an algorithm that selects prices (from the price set) throughout the experiment to maximize profits. The algorithms considered belong to a class known as multi-armed bandits (MABs), described in Section 4.

[3] The population of consumers has a population-level distribution of WTP, which corresponds directly to the demand curve.

We consider a few additional assumptions in this pricing problem. First, consumers are randomly drawn from a population with a stable WTP distribution of valuations. Second, consumers are short-lived, and the overall distribution of price expectations remains unaffected by the experimentation. These assumptions are typical in field experiments and necessary for the results to be applicable once the experiment has ended.[4] Third, we assume the firm operates as a single-product monopolist.[5] Fourth, the firm seeks to set a single optimal price, meaning it does not engage in price discrimination and does not consider inventory or other factors (Besbes and Zeevi 2009, Ferreira et al. 2018, Ringbeck and Huchzermeier 2019, Misra et al. 2019).[6] Finally, given the pace at which prices must be adjusted for efficient experimentation, our algorithms are suited to the online domain rather than physical stores.

3.2. Simple Pricing Experiment Example

We now illustrate a simple example of a balanced pricing experiment with certain valuation distributions to show why existing bandit algorithms may struggle, and how our algorithm can enhance performance. Consider a monopolist firm selling a single-unit product of zero marginal cost with an unknown demand curve $D(p) = 1 - p$, testing prices tested $P = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ each 100 times. Figure 1 presents the results of demand and profit learning. Figure 1a) shows sample mean demand at tested prices (blue dots) with 95% credible intervals, while Figure 1b) displays sample mean profit (red dots) with 95% credible intervals. Grey dotted lines represent the true demand and profit curves.

This figure highlights a key phenomenon: the credible intervals for demand are fairly consistent,[7] but those for profit vary greatly and expand with price due to the scaling effect. Learning occurs at the demand level, but optimal pricing decisions depend on profit.

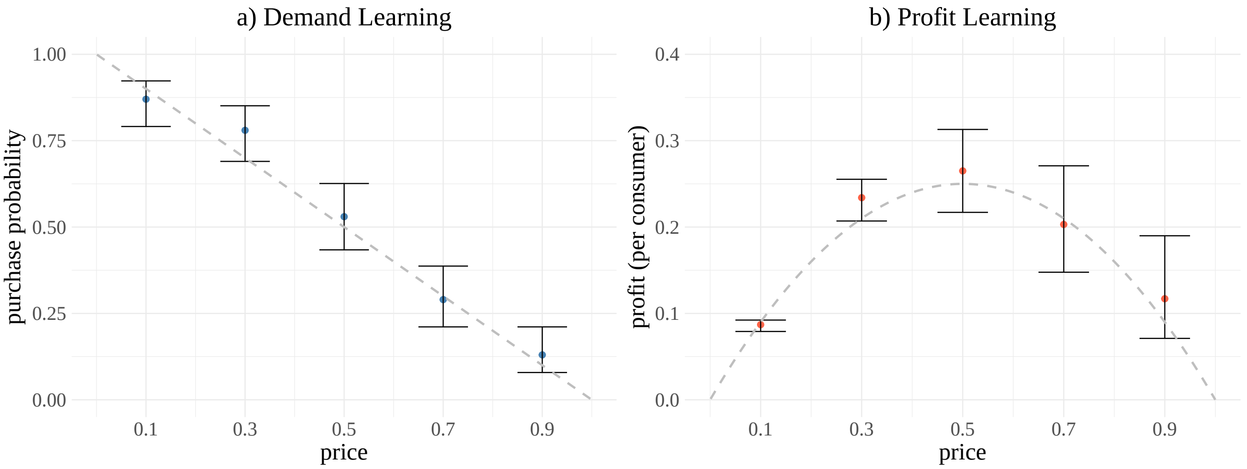
[4] This excludes dynamic situations where consumers may change over time or where current decisions are heavily influenced by future expectations. Such cases include strategic consumers (Nair 2007), learning (Erdem and Keane 1996, Yu et al. 2016), and stockpiling (Ching and Osborne 2020, Hendel and Nevo 2006). These assumptions are often implicit in field experiments, such as in the advertising literature (Hoban and Bucklin 2015, Lambrecht et al. 2018, Gordon et al. 2019). For instance, if strategic consumers believe that a firm offering discounts is experimenting and may discount further later, the observed treatment effect may not accurately reflect reality.

[5] This assumption can be relaxed; the results will hold as long as the algorithm is deployed in a stable environment where (1) competitors do not change prices strategically in response to real-time changes and (2) firms are unconcerned with potential future competitors (Rubel 2013).

[6] Inventory constraints are relevant when stock is limited, as in clothing. When products are constrained, it may be preferable to forgo selling to one consumer to sell to another with a higher WTP. For products without production constraints, such as a Netflix subscription, this issue does not arise.

[7] Small differences are discussed in Appendix EC.6.

Figure 1 Demand and Profit Learning for a Balanced Experiment



Notes. Results of a single balanced experiment where each of the prices $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ was tested 100 times with a true unknown demand of $D(p) = 1 - p$. Figure a) shows the mean and 95% credible intervals for purchase probability at each price tested – the dotted grey line shows the true purchase probability. Figure b) shows the mean and 95% credible intervals for profit at each price tested – the dotted grey line shows the true expected profit.

Consequently, the difficulty of identifying the optimal price depends on its position within the tested prices. A high optimal price simplifies the task, as learning that lower prices yield smaller profit intervals is less challenging. Conversely, a low optimal price complicates the task, as determining that higher prices are suboptimal demands more consumer purchase data. This results in poorer algorithmic performance when the optimal price is low. Incorporating monotonicity enhances performance by excluding any demand curve where demand increases with price, thereby narrowing the range of possible demand curves and reducing demand intervals. This effect is amplified by the scaling effect, significantly lowering the profit intervals at high prices.

4. Model Preliminaries

4.1. MAB Components

This section formally introduces multi-armed bandits (MABs) and their application to the pricing problem. Notably, all MABs have three components: actions, rewards, and a policy.

The first component – *actions* – refers to the set of prices from which the firm can choose. Prior to the experiment, the firm selects a finite set of A ordered prices $P = \{p_1, \dots, p_A\}$, where $p_1 < p_2 < \cdots < p_A$, and prices are scaled so that $0 \leq p_a \leq 1$.[8] While this paper does not explicitly model how to choose the set of prices, the general trade-off is that learning is easier with fewer prices, but a higher optimum is possible when more prices are considered.

[8] With unscaled prices $\{\widetilde{p_1}, \dots, \widetilde{p_A}\}$, the set of scaled prices can be created by dividing any price by the largest price in the set, i.e., $p_a = \widetilde{p_a}/\widetilde{p_A}$.

Table 1 Summary of Bandit Notation

Notation	Description	Formula
Ψ	Pricing policy (i.e., decision rule)	Depends on algorithm
a	Action from the set of actions $\mathcal{A}$	$a \in \mathcal{A} = \{1, 2, ..., A\}$
t	Time-step: denotes the t-th consumer of the price experiment	
P	Set of prices to be tested	
p_a	Scaled price corresponding to action a	$p_a \in P = \{p_1, ..., p_A\}$ where $0 \leq p_a \leq 1 \; \forall p_a$
n_{at}	Number of times price p_a has been chosen through time t	
s_{at}	Number of purchases at price p_a through time t	
H_t	History from past t rounds of experiment	$H_t = \{S_t = (s_{1t}, ..., s_{At}), N_t = (n_{1t}, ..., n_{At})\}$
a_t	Action chosen at time t	$a_t = \Psi(P, H_{t-1})$
$D(p_a)$	Demand at price p_a	
y_{at}	Purchase rate through time t of price p_a	$y_{at} = s_{at}/n_{at}$
$\bar{\pi}_{at}$	Mean profit through time t of price p_a	$\bar{\pi}_{at} = p_a(s_{at}/n_{at})$
π_{a_t}	Profit realized when price p_a was tested in time period t	

The results section provides general guidelines for how to pick the set of prices. Once P is chosen, at each time-step t of the experiment, the firm chooses a price p_a from P.

The second component – *rewards* – refers to the profits that a firm makes at each purchase opportunity. The firm faces an unknown true demand $D(p)$, and the true profit function is given by $\pi(p) = pD(p)$. We assume variable costs are zero, though the model can easily accommodate such costs.[9] The true profit is not observed; instead, the firm observes noisy realizations of profits corresponding to each price p_a. Considering the data at each price separately, we define n_{at} to be the number of times that price p_a has been chosen through time t, and s_{at} to be the cumulative number of purchases for action a through time t. The observed purchase rate through time t for price p_a is simply $y_{at} = \frac{s_{at}}{n_{at}}$. Accordingly, the mean profit for a price p_a at time t is $\bar{\pi}_{at} = p_a\left(\frac{s_{at}}{n_{at}}\right)$.

The final component – a *policy* – denoted by Ψ, is a decision-making rule that picks an action or price in each round using the history from past rounds. In this situation, the history can be written as $H_t = \{S_t = (s_{1t}, \ldots, s_{At}), N_t = (n_{1t}, \ldots, n_{At})\}$. Formally, in round t, the policy picks a price using the history from the past $(t-1)$ rounds: $p_{a_t} = \Psi(P, H_{t-1})$. What distinguishes various MAB algorithms is how this policy is defined. For a typical randomized experiment, the policy can be defined as an equal probability across all arms, completely ignoring history. A summary of the notation is given in Table 1.

[9] Simply, the reward obtained from a potential consumer who purchases is the price charged minus the cost.

4.2. Performance Metrics

To assess the performance of our proposed algorithm, we need to compare it to other bandit algorithms. This is equivalent to a comparison of policies, as only the policy Ψ depends on the algorithm, while the price set and arm rewards are common across algorithms.

Among the various performance metrics in the bandit literature, the most common is regret, which is defined as the difference between rewards under full knowledge (always playing the optimal price) and the expected rewards from the policy in question (Lai and Robbins 1985). Formally, the cumulative regret of policy Ψ through time t is

$$\text{Regret}(\Psi, P, t) = \mathbb{E}\left[\sum_{\tau=1}^{t}(\pi^* - \pi_{a_\tau} \mid \Psi, P, H_{\tau-1})\right] \tag{1}$$

where P is the set of prices being considered, π^* is the ex-post maximum expected profit in a given round, and π_{a_τ} is the profit realized when price p_a is played in time period τ. This metric[10] is used in the discussion of theoretical properties in Appendix EC.4.[11]

An alternative objective is to maximize the expected total reward (Gittins 1974, Cohen and Treetanthiploet 2020).[12] Formally, the goal is to pick a decision rule, Ψ, that selects a sequence of prices from the consideration set, P, that maximizes the total expected profit:

$$\mathbb{E}\left[\sum_{\tau=1}^{t}\pi_{a_t} \mid P\right] \tag{2}$$

We use this metric to discuss our empirical simulation results, as it is more intuitive in the context of the pricing problem (a firm maximizing profits through experimentation with an unknown demand curve). Additionally, if the true rewards distribution is unknown (e.g., in a field experiment), total rewards can still be compared, whereas regret lacks a straightforward alternative formulation.

[10] Formulating the bandit problem as a statistical problem (regret) rather than an optimization problem (maximizing cumulative reward) lends itself better to theoretical guarantees (Cohen and Treetanthiploet 2020).

[11] Theoretical guarantees often state that an algorithm has the lowest possible bound for expected regret; however, this is subtly different from empirical performance, which may be higher for algorithms without such theoretical properties. For example, it is proven that under certain conditions, UCB has the lowest possible bound for expected regret (Auer et al. 2002). However, empirically, under the same conditions, it is often outperformed by greedy algorithms with respect to maximizing rewards (Bayati et al. 2020).

[12] Among a set of algorithms, the one with the lowest cumulative regret is the same as the one with the highest cumulative rewards.

4.3. Baseline Policies – Deterministic and Stochastic Algorithms

Finally, we consider baseline policies, which serve as building blocks for our algorithm. The simplest approach is a fully randomized experiment (A/B testing), where a random arm is selected, ignoring the history of arms played and their outcomes. Another class includes myopic policies, such as greedy-based algorithms (e.g., ϵ-greedy) and softmax (Dann et al. 2022). In this paper, we build on two popular policies, UCB and TS, which tradeoff learning and earning, and form a fundamental component of typical bandit algorithms.

UCB: The Upper Confidence Bound (UCB) algorithm is a deterministic, nonparametric approach popularized for its proven asymptotic performance, achieving the lowest possible maximum regret (Lai and Robbins 1985, Agrawal 1995, Auer et al. 2002). The UCB policy scores each arm by summing an exploitation term and an exploration term, then selecting the arm with the highest score. The exploitation term is the sample mean of past rewards at a given arm, which provides information about past payoffs. The exploration term, meanwhile, increases with the uncertainty of the sample mean for an arm; specifically, it decreases as an arm is chosen more frequently and as the empirical variance of rewards at that arm decreases. Thus, the UCB policy balances exploitation and exploration.[13] To adapt UCB to pricing, we scale the exploration term by price, as shown in eq. (3) (Misra et al. 2019). This adjustment accounts for the known variation in the range of possible rewards at each arm, which is not generally assumed.

$$\begin{aligned} p_a^{\text{UCB}} &= \underset{p_a \in P}{\arg\max} \left(\bar{\pi}_{at} + p_a \sqrt{\frac{\log(t)}{n_{at}} \min\left(\frac{1}{4}, V_{at}\right)} \right) \\ V_{at} &= \left(\frac{1}{n_{at}} \sum_{\tau=1}^{n_{at}} \pi_{a\tau}^2 \right) - \bar{\pi}_{at}^2 + \sqrt{\frac{2\log t}{n_{at}}} \end{aligned} \tag{3}$$

TS: Thompson Sampling (TS) is a randomized Bayesian parametric approach. For each arm, a reward distribution is specified with a prior, and updated based on the history of past trials (Thompson 1933). In each round, an arm is chosen according to the probability that it is optimal, given the history of past trials. Specifically,

$$\text{Prob}(p_a \mid H_{t-1}) = \text{Prob}(E[\pi_{a,t} \mid p_a] > E[\pi_{a,t} \mid p_{a'}], \forall p_{a'} \neq p_a \mid H_{t-1}). \tag{4}$$

The simplest implementation is to sample from each distribution in each round and select the arm with the highest sampled payoff. In our setting, where purchase decisions are

[13] If the exploration term were zero, this would be equivalent to a fully greedy algorithm.

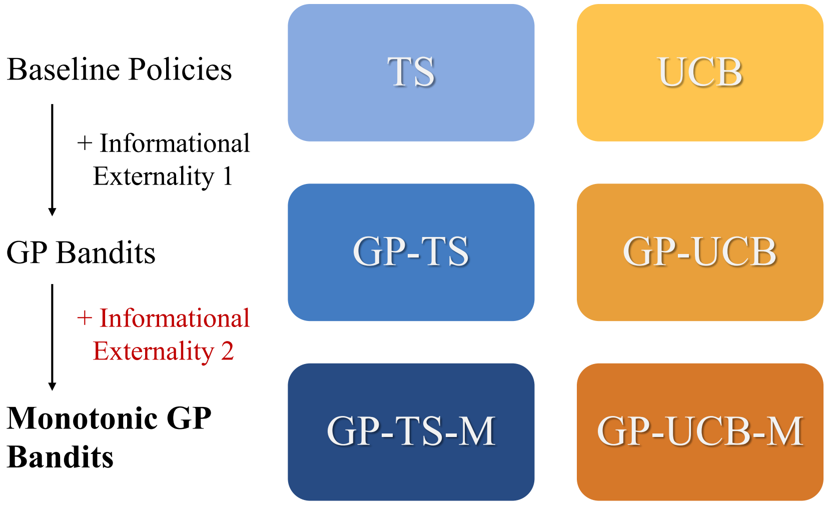
binary, the TS approach uses a scaled beta distribution, with parameters representing the number of successes and failures (Chapelle and Li 2011, Agrawal and Goyal 2012). Specifically, at time $t+1$ for each arm a, a sample is drawn from Beta$(s_{at}+1, n_{at}-s_{at}+1)$ and then scaled by the price, p_a. The arm with the highest value is then chosen.

Another benchmark we consider is Nonparametric Thompson Sampling (N-TS) by Urteaga and Wiggins (2022), which extends standard TS to handle reward model uncertainty. Unlike traditional TS, which requires knowledge of the true reward model, N-TS uses Bayesian nonparametric Gaussian mixture models to flexibly estimate each arm's reward distribution. This approach allows the complexity of the per-arm reward model to adjust dynamically as new observations are collected, enabling sequential learning and improved decision-making.

5. Informational Externalities

In this section, we incorporate the two informational externalities into the baseline policies. Our objective is to create a general method that combines any decision rule (such as UCB, TS, or others) with the two relevant informational externalities. The first informational externality – continuity – is implemented using Gaussian process bandits developed by Srinivas et al. (2009). The main contribution of this paper, however, is the incorporation of the second externality – monotonicity – which also builds on the Gaussian process framework. An overview of how these externalities are incorporated into the baseline policies is shown in Figure 2.

Figure 2 Overview of Incorporation of Informational Externalities



5.1. First Informational Externality: From Points to Functions

The first informational externality recognizes the local dependence of functions through continuity, whereas baseline MABs consider outcomes at discrete arms to be separate and independent. Specifically, in the pricing application, we know that demands at two prices tend to be closer when the prices are closer together. With 10 arms, to inform demand at arm a (say price $p_a = 0.5$), the demands at $p_{a-1} = 0.4$ and $p_{a+1} = 0.6$ are most informative. More generally, it is possible to learn about the demand $D(p)$ at price p from observed demands at nearby price points, such as $D(p + \epsilon)$ for small ϵ. Note that the information spillover is bidirectional, with points further away being less important and weighted less due to the structure of the covariance matrix.

The logic of sharing information between arms means that using functions to model demand across the range of prices, rather than focusing only on demand at specific arms (price points), may lead to increased performance. A straightforward parametric approach would involve specifying functional forms (e.g., splines) to flexibly model the true demand curve, using observations at the arms (prices). However, there is a risk that any parametric approximation chosen by the researcher may be insufficient to capture the true shape of the demand curve.

We take a more flexible nonparametric approach by modeling the space of demand functions as a Gaussian process (GP). A Gaussian process is a stochastic process (a collection of random variables) such that every finite subset has a multivariate Gaussian distribution; a simple example of fitting a GP to data can be found in Appendix EC.1. GPs can be thought of as a *probability distribution over possible functions*, allowing any function to be probabilistically drawn from the function space on the chosen support, unlike typical parametric models. Thus, any arbitrary demand curve can be modeled, and the GP learns the shape from the data. Following Srinivas et al. (2009), GPs can be incorporated into the bandit framework using both UCB and TS (Chowdhury and Gopalan 2017).

Advantages of GPs relative to other methods: GPs offer several desirable features for the present class of problems. First, GPs provide a parsimonious, nonparametric approach to incorporating both informational externalities in a transparent, principled, and provable manner.[14] Second, GPs have closed-form solutions that allow hyperparameters to be

[14] An alternative approach would be to use a parametric model like *GLM-UCB* (Filippi et al. 2010) and restrict the coefficients to obtain weakly decreasing demand functions.

tuned quickly with maximum likelihood estimation. Intuitively, GPs work well with bandits because the exploration-exploitation trade-off relies on understanding the complete distribution of rewards, not just the mean reward at each arm.[15] A GP, specifying probability distribution over functions, offers a principled way to manage the exploration-exploitation trade-off.

Finally, we note that other nonparametric methods, including many machine learning methods, may provide higher-accuracy estimates for the mean. However, because they are not typically capable of quantifying the certainty of their predictions, they are ill-suited for bandit algorithms, where managing the exploration-exploitation trade-off is crucial. In contrast, GPs directly quantify uncertainty, allowing it to be incorporated into the decision-making process in a principled way. The value of modeling uncertainty has been demonstrated by ignoring uncertainty using only the means of the posterior GP rather than the entire posterior.[16]

5.1.1. Gaussian Processes The key building block of our approach, which underpins the modeling of both informational externalities, is the Gaussian process. As in Ringbeck and Huchzermeier (2019), we model the GP at the demand-level and then scale by price, allowing the bandit to make decisions at the reward-level.

Table 2 Summary of GP Notation

Description	Notation in Pricing Setting
Training data	$\mathcal{D}_t = \{P_t, y_t\}$
Noise hyperparameter	σ_y^2
RBF kernel	$k^{\mathrm{RBF}}(p_i, p_j) = \sigma_D^2 e^{\frac{-(p_i-p_j)^2}{2l^2}}$
RBF hyperparameters	$\{\sigma_D, l\}$
Covariance function (kernel) evaluated at two points	$k(p_i, p_j)$
Covariance matrix between price vectors	$K(P, P)$

Training Data: The goal of estimating a GP here is to learn the space of demand functions at test points, P,[17] given the purchase decision history obtained at time t from the MAB experiment. The training data is defined as $\mathcal{D}_t = \{P_t, y_t\}$, where $P_t = \{p_a : a \in \mathcal{A}_t\}$ and

[15] This is also incorporated to some degree in the UCB algorithm, which models both a term for the sample mean of each arm and an exploration bonus dependent on the bounds surrounding those sample means.

[16] This approach can be thought of as a greedy-based GP algorithm. This was tested in Srinivas et al. (2009), who found it was "too greedy too soon and tends to get stuck in shallow local optima" (p. 4), leading the algorithm to under-explore and produce inaccurate results.

[17] In our setup, we use only the prices from the test set, though it is possible to estimate for any arbitrary price.

$y_t = \{y_{at} : a \in \mathcal{A}_t\}$. Here, $\mathcal{A}_t \subseteq \mathcal{A}$ is the subset of actions tested by time t, and $y_{at} = \frac{s_{at}}{n_{at}}$ represents the observed purchase rate for price p_a at time t. The purchase rates, y_{at}, provide a noisy signal of the true value of the demand function $D(p)$ at p_a; specifically, we assume $y_{at} = D(p_a) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_{y_a}^2)$. The vector $\sigma_y^2 = (\sigma_{y_1}^2, \ldots, \sigma_{y_A}^2)$ is referred to as the noise hyperparameter. Importantly, the cardinality of the training data depends on the number of prices in the test set, $|P| = A$. Therefore, it does not depend on t, the number of purchase decisions observed, reducing the dimensionality. The computation and equivalence of these two approaches is discussed in Appendix EC.2.

Kernel: A Gaussian process flexibly models dependencies across input points using a kernel, implemented through a covariance function. A kernel $k(\cdot,\cdot)$ takes two points (in our setting, prices p_i and p_j) from the input space and returns a scalar representing the covariance between the outputs at those points. From this, a covariance matrix $K(\cdot,\cdot)$ can be constructed for a set of inputs.

A commonly used and robust kernel for GPs is the radial basis function (RBF) kernel, also known as the Gaussian kernel or the squared exponential kernel (Duvenaud 2014).

$$k^{\mathrm{RBF}}(p_i, p_j) = \sigma_D^2 e^{-\frac{(p_i - p_j)^2}{2l^2}} \tag{5}$$

The RBF kernel has a set of desirable properties suited to our setting. It is differentiable and can provably approximate any arbitrary continuous target function uniformly on any compact subset of the input space (Micchelli et al. 2006).

Shape Hyperparameters: The RBF kernel has two shape hyperparameters, σ_D and l. The first hyperparameter, σ_D, is a scale factor that controls the amplitude of the function (i.e., the average distance of the function from its mean). The second hyperparameter, l, determines the smoothness[18] of the function, describing how the correlation between two points decreases as the distance between them increases (Shahriari et al. 2015).

A crucial practical step in estimating the GP is tuning the hyperparameters. To minimize human (researcher) judgment, it is ideal to have an efficient, accurate, and automatic method for selecting hyperparameters. We use standard non-Bayesian methods for tuning the hyperparameters.[19] This approach involves choosing values of the hyperparameters that maximize the likelihood of the data given the model. Mathematically, this is equivalent to

[18] Intuitively, this can be thought of as the length of the “wiggles.”

[19] Bayesian tuning methods are often too slow and not suitable for real-time bandit settings.

minimizing the negative log marginal likelihood (equation 2.30 in Williams and Rasmussen (2006)):

$$\log \text{prob}(y_t|P_t) = -\frac{1}{2}y_t^T(K(P_t,P_t)+\sigma_y^2 I)^{-1}y_t - \frac{1}{2}\log|K(P_t,P_t)+\sigma_y^2 I| - \frac{t}{2}\log(2\pi) \tag{6}$$

Noise Hyperparameter: While the noise hyperparameter can be estimated along with the shape hyperparameters, we choose to specify it directly due to the difficulty of disentangling shape and noise hyperparameters (Murray 2008).[20] A conservative approach to specifying the noise hyperparameter –without estimation – is to use the upper bound. We leverage the fact that purchase decisions in our model follow a Bernoulli distribution and lie within the interval $[0,1]$ to calculate this upper bound, which is 0.25 and occurs when the true purchase probability is 0.5.[21] While this may be overly conservative for some prices, in contrast, setting noise hyperparameters too low could limit exploration and lead to poor results by causing the algorithm to get stuck in errant equilibria. A discussion of alternative methods, where noise takes on different values for different prices (i.e., heteroscedastic noise), is provided in Appendix EC.6.

Posterior Prediction: Estimating the posterior GP is done using standard GP regression on training data. Equations and an illustrative example are provided in Appendix EC.1.

5.1.2. GP-UCB and GP-TS Gaussian processes can be combined with bandits, such as UCB and TS, to create *GP-UCB* (Srinivas et al. 2009) and *GP-TS* (Chowdhury and Gopalan 2017), respectively. The general idea is that instead of using the raw data directly, a posterior GP is estimated before applying a UCB or TS rule to choose a price. As with the baseline algorithms, we scale by price to adapt to the pricing setting.

To initialize the algorithm, we select the first price randomly.[22] Once one data point is available, the training data $\mathcal{D}_t = \{P_t, y_t\}$ can be used to choose the hyperparameters using equation (6). Additionally, the posterior mean $\mu(D^*)$ and covariance matrix $\text{Cov}(D^*)$ can be calculated using equations (EC.5) and (EC.6).[23]

[20] For example, consider two close input points (x-axis) that have very different outputs (y-axis). One possible explanation is that the data is accurate, and the GP requires shape parameters that permit sufficiently high variation to capture large output differences from nearby inputs. Alternatively, the true outputs may be close together, but the data is very noisy; in this case, the previous shape parameters would be overfitting.

[21] As we use purchase rates rather than purchase decisions as the training data, we also need to divide by n_{at} at each price. See Appendix EC.2 for details.

[22] Another possible initialization method is to set arbitrary hyperparameters and model the GP without data. Both methods are practical, with only minor differences in overall performance (under 1% in all our simulations), and neither method consistently outperforms the other.

[23] D^* is a random variable denoting the Gaussian process posterior prediction.

With the GP estimated, we can apply either UCB and TS to select a price arm. For GP-UCB, at each price $p_a \in P$, the posterior demand mean $\mu_t(p_a)$ and variance $\sigma_t^2(p_a)$ are used to determine the price at iteration $t+1$ according to the following decision rule:

$$p_a^{\text{GP-UCB}} = \arg\max_{p_a \in P} \left(p_a \left(\mu_t(p_a) + \beta_{t+1}^{1/2} \sigma_t(p_a) \right) \right) \tag{7}$$

where $\beta_t = \frac{2}{5} \log(|P| t^2 \pi^2 / (6\delta))$.[24] Note that we have scaled by p_a as the algorithm is optimizing for reward rather than demand (the level at which the GP was estimated).

On the other hand, in GP-TS, rather than sampling at each arm as in traditional TS, a demand draw for every test price, $D^{(t)}(p_a)$, can be obtained by sampling from the posterior GP. Specifically, $D^{(t)}(P)$ is a sample from the posterior normal distribution with the given mean and covariance matrix, $D^* \sim N(\mu(D^*), \mathrm{Cov}(D^*))$. Then, using Thompson sampling, the selected price will be

$$p_a^{\text{GP-TS}} = \arg\max_{p_a \in P} \left(p_a D^{(t)}(p_a) \right) \tag{8}$$

5.2. Second Informational Externality: Monotonicity

We now focus on the main contribution of this paper: incorporating the second informational externality into Gaussian process bandits. The monotonicity property has a global influence, as demand at price p_j, $D(p_j)$, constrains all demands at higher prices since $D(p_i) \leq D(p_j)$ when $p_i \geq p_j$. Similarly, all demands at lower prices are constrained as well: $D(p_i) \geq D(p_j)$ when $p_i \leq p_j$ Note that the impact is asymmetric: specifically, demand at a higher price p_i is upper-bounded by demand at a lower price p_j, while demand at a lower price is lower-bounded by demand at a higher price.

Our method is general enough to integrate with both GP-UCB and GP-TS, as past literature (Chowdhury and Gopalan 2017) shows that neither completely dominates the other. We refer to the monotonic versions of GP-UCB and GP-TS as *GP-UCB-M* and *GP-TS-M*, respectively. The goal is to determine whether and under what conditions incorporating monotonicity improves performance for either variant.

When using GP-TS or GP-UCB, the baseline GP allows for any demand function and does not impose any restrictions on the shape. Our goal is to obtain only weakly decreasing monotonic demand functions. For GP-TS-M, we require a way to randomly draw a

[24] Srinivas et al. (2009) found in their experiments that $\delta = 0.1$ works well empirically, and we use that value here. Other values of β may perform better in different simulations, though determining β without past data is generally challenging (Hoffman et al. 2011).

monotonic function from the set of monotonic functions in the posterior GP. For GP-UCB-M, we require an estimate of the mean and variance from the subset of monotonic demand curves from the posterior; alternatively, this can be approximated by averaging over multiple monotonic draws.

To obtain a random monotonic draw, a simple approach is to use rejection sampling by repeatedly sampling from the GP until a weakly decreasing draw is obtained. While this approach can work, there is no guarantee that a weakly decreasing draw will be found quickly. This issue is further exacerbated when there are few observations or when there are many test prices; faced with many arms and (noisy) non-monotonic sample means, the probability of finding a monotonic draw can become vanishingly small.

To ensure that a weakly decreasing draw can be obtained from a GP expediently in all cases, we develop a method from first principles. We note that sampling a monotonic function from a GP is intractable. We therefore transform the problem by leveraging a helpful property of GPs that the derivative of a GP is also a GP (and that the RBF kernel is infinitely differentiable), allowing us to estimate the derivative of the GP from our data. The transformed problem, where we obtain a draw from a GP with all values being negative, then becomes a tractable sampling problem (equivalent to sampling from a truncated normal). Specifically, a decreasing monotonic function can be characterized by having negative first derivatives at all points. We then use the basis functions proposed by Maatouk and Bay (2017) to estimate the demand function from a random sample of negative derivatives.

An important property of this principled approach is that the function is guaranteed to be monotonic not only at the discrete price levels forming the support but also at any intermediate price where no experimentation is performed. This ensures consistency in the function's behavior across the entire price range. The only assumption required is that the demand function is differentiable with a continuous derivative.

5.2.1. Basis Functions To estimate the demand function, we use a collection of functions h_j known as the interpolation basis. These basis functions are defined a priori and remain the same regardless of the input data. They provide a method for estimating a function at all points by linearly interpolating between known function values at knots spaced over the support. Following Maatouk and Bay (2017), let $u_j \in [0,1]$, for $j = 0,1,\ldots,J$, denote equally spaced knots on $[0,1]$ with spacing $\delta_J = 1/J$ and $u_j = j/J$. The interpolation basis is defined as

$$h_j(p) = h\left(\frac{p-u_j}{\delta_J}\right) \text{ where } h(p) = (1-|p|)\mathbb{1}(p \in [-1,1]). \tag{9}$$

Then, for any continuous function $D:[0,1] \to \mathbb{R}$, the function

$$D_J(\cdot) \approx \sum_{j=0}^{J} D(u_j) h_j(\cdot) \tag{10}$$

approximates D by linearly interpolating between function values at the knots u_j. A key property of the interpolation basis is that, as the gap between the evenly spaced knots becomes infinitesimally small, the distance between the estimate and the true function converges to 0.

Our goal, however, is to simplify the sampling problem by approximating the demand function in terms of its derivatives at the knot points. Equation 10 can be transformed to express the demand function in terms of its intercept, derivatives, and the original basis functions h_j, as shown in Proposition 1 (a derivation is provided in Appendix EC.3.2). Appendix EC.3.1 provides a visual demonstration of the basis functions h_j and their integrals $\int_0^p h_j(x)\,dx$.

Proposition 1. *Assuming a demand function $D:[0,1] \to \mathbb{R}$ is differentiable with a continuous derivative (i.e., $D \in C^1([0,1])$), it can be estimated by its intercept and derivatives using the following equation:*

$$D(p) \approx D(0) + \sum_{j=0}^{J} D'(u_j) \int_0^p h_j(x)dx \tag{11}$$

While this approach applies to all class C^1 functions on the support, we additionally assume that the unknown demand function D is weakly decreasing, meaning that it belongs to a subset $\mathcal{M}$ defined as follows:

$$\mathcal{M} := \{D \in C^1([0,1]) : D'(p) \leq 0, p \in (0,1)\} \tag{12}$$

In other words, D belongs to the subset of functions where the derivative is never positive at any value of p.

5.2.2. GP-UCB-M and GP-TS-M There are a few differences between the monotonic and non-monotonic versions of GP bandits. When incorporating monotonicity, instead of estimating a GP of the means at the test prices, we first estimate a GP of the derivatives at the knots, concatenated with the mean at the intercept. Crucially, the intercept and derivatives must be estimated together, which is possible due to the property of a GP that the joint distribution of values and their derivatives is also a GP (Appendix EC.3.3 provides details on calculating the derivative of a GP). Once the posterior GP is estimated, off-the-shelf sampling can be used to acquire a draw where every derivative is non-negative, specifically we use the *TruncatedNormal* package in R (Botev and Belzile 2021). Then, equation (11) provides a formula to recover the demand sample $D^{(t)}$ at the desired test prices using only the draw and basis functions h. From this point, the decision rules for GP-UCB or GP-TS can be applied as usual. Formally, the methods are outlined in Algorithms 1 and 2.

Algorithm 1: GP-TS-M

```
Set test prices $P$, kernel $k$, noise hyperparameter $\sigma_y^2$, and knots $\mathcal{U}$
Compute the integrals of the basis functions at $\mathcal{U}$ using equation (9)
Define test points as $\mathcal{U}_0 = \{0\} \cup \mathcal{U}$ (concatenate the intercept)
For $t = 1$ pick price randomly, and observe purchase decision
Initialize training input $P_t$ and training output $y_t$
for $t = 2, 3, \ldots$ do
    Compute shape hyperparameters $\sigma_D$ and $l$ using equation (6)
    Compute covariance matrix (equation (EC.3)) using $P_t$ and $\mathcal{U}_0$ with equations (EC.11), (EC.12),
      (EC.13)
    Estimate posterior GP using equations (EC.5) and (EC.6)
    Sample randomly from the posterior GP at test points $\mathcal{U}_0$
    Estimate the demand draw $D^{(t)}$ at test prices $P$ using equation (11)
    Play price $p_a = \arg\max_{p_a \in P} \left(p_a D^{(t)}(p_a)\right)$
    Observe purchase decision
    Update $P_t$ and $y_t$
end
```

5.2.3. Theoretical Properties We now characterize the theoretical properties of our algorithms.[25] Our main theoretical result shows a regret with a main term that scales like

[25] We make a slight adjustment by relaxing the monotonicity assumption, requiring monotonicity only at the knot points rather than across all points

```
Algorithm 2: GP-UCB-M
Set test prices $P$, kernel $k$, noise hyperparameter $\sigma_y^2$, and knots $\mathcal{U}$
Compute the integrals of the basis functions at $\mathcal{U}$ using equation (9)
Define test points as $\mathcal{U}_0 = \{0\} \cup \mathcal{U}$ (concatenate the intercept)
For $t = 1$ pick price randomly, and observe purchase decision
Initialize training input $P_t$ and training output $y_t$
for $t = 2, 3, \dots$ do
    Compute shape hyperparameters $\sigma_D$ and $l$ using equation (6)
    Compute covariance matrix (equation (EC.3)) using $P_t$ and $\mathcal{U}_0$ with equations (EC.11), (EC.12), (EC.13)
    Estimate posterior GP using equations (EC.5) and (EC.6)
    Obtain $B$ samples from the posterior GP at test points $\mathcal{U}_0$
    Estimate the demand draw $D^{(t,b)}$ at test prices $P$ using equation (11) for each of the $B$ samples
    Estimate $\mu_t(p_a)$ and $\sigma_t(p_a)$ from the collection of demand draws
    Play price $p_a = \arg\max_{p_a \in P} \left(p_a \left(\mu_t(p_a) + \beta_{t+1}^{1/2} \sigma_t(p_a)\right)\right)$
    Observe purchase decision
    Update $P_t$ and $y_t$
end
```

$O\left(\mathbb{E}\left[\sqrt{\gamma_T \sum_{t=1}^T p_t^2}\right]\right)$ where γ_T is a problem dependent quantity reflecting the underlying effective dimension of the kernel, and p_t is the price played at time t. Our analysis follows the standard approach of Thompson sampling for Gaussian processes; we refer the reader to Srinivas et al. (2009) for more details on the interpretation of γ_T. We note that, in the case of the RBF kernel, γ_T is $\log(T)$, and therefore has a negligible contribution to the regret. As we discuss in Appendix EC.4, the regret also consists of a smaller term E_T, which we show empirically to be bounded by a sublinear term.

The most interesting aspect of the regret is the improved "path-dependent" analysis, which reflects the actual prices played and demonstrates the advantage of exploiting the monotonic structure (Zhao et al. 2023). At worst, if we naively upper bound p_t by 1, this yields a regret of $O(\sqrt{\gamma_T T})$. However, this expression does not capture the true behavior of our proposed method. Intuitively, since we are achieving sub-linear regret, we expect that, as time progresses, the price p_t converges to p^* as $t \to \infty$. Thus, for large T, our regret is closer to $p^*\sqrt{T} \leq \sqrt{T}$.

Furthermore, as we discuss in Section 3.1, by multiplying the demand by p to obtain profits, we are effectively creating larger confidence intervals (and thus greater uncertainty) for profit at higher prices. In contrast, lower prices yield smaller confidence intervals and less uncertainty on the profit curve. Consequently, p_t may be very large in the earlier rounds as the algorithm strives to reduce uncertainty on the profit at higher prices, regardless of whether we use UCB or Thompson sampling. By enforcing monotonicity, this effect is still present but mitigated, as $D(p)$ is necessarily smaller for larger values of p, even if we have not fully learned the function at those values. As a result, the p_t we ultimately play should be smaller with monotonicity.

6. Analysis and Results

We now explain the implementation of the algorithms, specifically detailing key components such as consumer valuations, the number of arms, and the sequence of events in the multi-armed bandit.

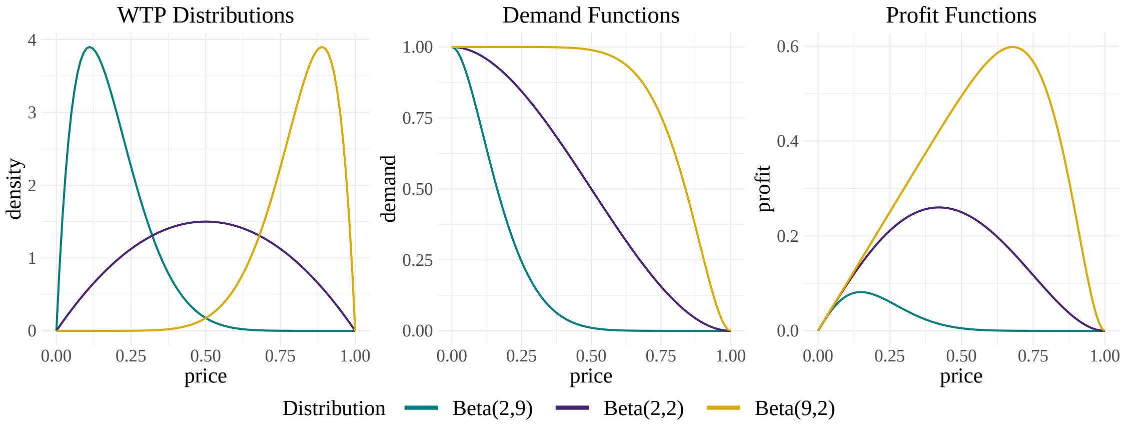
Sequence of Events: We evaluate our proposed algorithm and benchmarks using simulations based on a standard setup for assessing bandit algorithms in pricing (Misra et al. 2019). Each simulation has the following structure. First, at each time period, a potential buyer (consumer) arrives from a large pool of consumers by being drawn from an unknown but stable WTP distribution. They are shown a price chosen by the algorithm (which has no specific knowledge about this consumer), and they decide to purchase one unit if and only if their valuation for the product is greater than the price shown. The outcome (purchase or no purchase) is observed by the algorithm, which then updates its history of observations. We allow for the algorithm to update its price every 10 consumers.[26]

Varying the Number of Arms: When choosing the number of arms (prices), the decision maker must balance two competing considerations: the granularity of the price set and the complexity of the learning problem. A smaller set of test prices faces the risk that the best price within the set is far from the true optimal price. Conversely, testing a larger set of prices increases the complexity of learning, slowing convergence and potentially leading to higher cumulative losses. We evaluate performance across different price sets, normalized from 0 to 1, using intervals corresponding to 5, 10 and 100 arms.

[26] While prices could be updated every period (i.e., for each consumer), this may be impractical in real-world settings. To better reflect industry practices, we change prices every 10 consumers, as in Misra et al. (2019).

Valuation (WTP) Distributions: Right-skewed, Left-skewed, and Symmetric: To evaluate the performance of various MAB policies, it is crucial to consider different shapes of consumer valuation (WTP) distributions. Following Misra et al. (2019), we analyze three types of distributions using specific parameterizations of the Beta distribution: Beta(2,9) for a right-skewed distribution, Beta(9,2) for a left-skewed distribution, and Beta(2,2) for a symmetric distribution. A graphical depiction of the willingness to pay, the demand curve, and the profit curve for each simulation setting is provided in Figure 3. A monopolist with perfect knowledge of the demand curve would set prices to maximize profit. The true optimal prices are 0.15 for Beta(2,9), 0.42 for Beta(2,2), and 0.68 for Beta(9,2).

Figure 3 WTP Distributions and Demand and Profit Functions



Algorithm Initialization: The algorithms are initialized with either a prior or by limited experimentation. For UCB, we assume a prior that encourages exploration by treating every untested price as if it has been tested once and resulted in a purchase.[27] In contrast, TS- and GP-variants can use uninformed priors, enabling price selection even without prior data. To make the comparison more consistent,[28] we have GP-variants select the first price randomly, after which the price can be chosen using the data-estimated GP.

6.1. Results

Our results present several variants within the two main classes of algorithms (TS and UCB). Figure 4 illustrates the cumulative performance of each algorithm over time (or

[27] An alternative approach is to test each price before applying the UCB policy; however, our initialization priors allow for the possibility of not testing every price, which can improve algorithmic performance.

[28] When faced with an uninformed prior, there are slight differences in the distribution of the first price chosen by the monotonic and non-monotonic algorithms. This can lead to minute differences in performance (under 1% and usually around 0%) depending on the underlying WTP distribution attributable solely to the first price decision, which we mitigate by choosing the first price randomly.

number of consumers), whereas Table 3 provides the exact values from this figure at 500 and 2500 consumers. Table 4 shows the uplift, or gains, from incorporating the two informational externalities relative to the baseline algorithms. Finally, Figure 5 presents a histogram of prices played, offering insight into the behavior of the different algorithms.

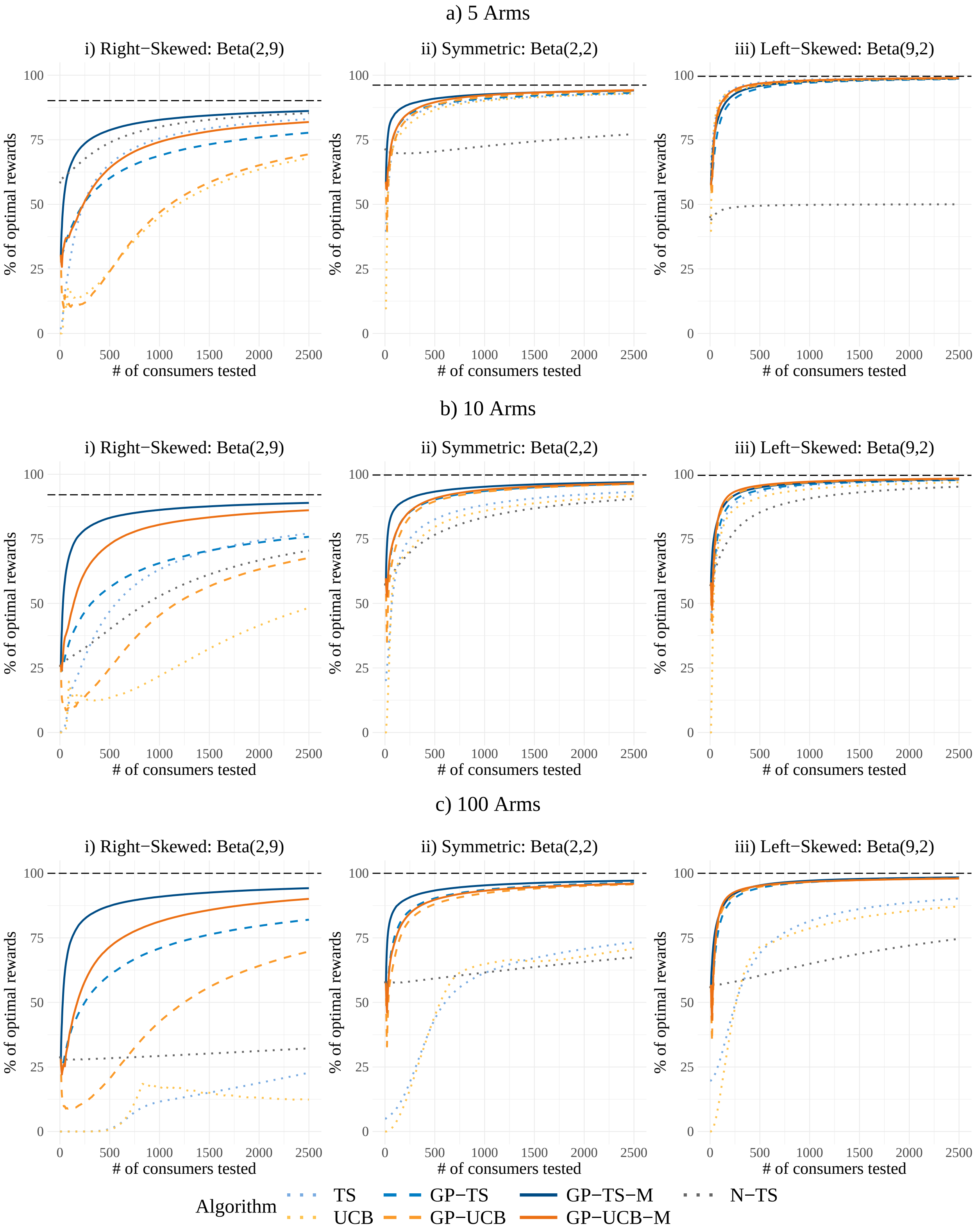
6.1.1. Main Results: Cumulative Performance The performances of the algorithms are shown in Figure 4, which reports the cumulative percentage of optimal rewards relative to the case when the optimal arm (price) is played.[29] In each subfigure, the horizontal black dotted line represents the maximum obtainable rewards given the price set (i.e., the ratio of the reward obtained from playing the best price within the price set to the true optimal reward). The value of this line increases as the price set enlarges, approaching the true optimal of 100%. Visually, algorithms with no informational externalities are represented with dots, those with the first informational externality (i.e., the GP) are represented with dashes, and those with both informational externalities are represented with solid lines. UCB-variants are shown in warm (orange-based) colors, while TS-variants are in cool (blue-based) colors.

In general, the results show that incorporating informational externalities improves algorithmic performance, though the effects vary across distributions and the number of arms. We explore this in more detail, starting with the case of 5 arms, illustrated in Figure 4a. For the right-skewed Beta(2,9) distribution, despite all algorithms eventually converging to the optimal price with sufficient learning, there is a significant separation in performance across algorithms even after 2500 consumers. Incorporating the first informational externality – a Gaussian process – results in minimal improvement for UCB and a slight decrease in performance for TS. However, incorporating the second informational externality – monotonicity – leads to substantial performance gains for both GP-UCB and GP-TS. Finally, the Nonparametric TS (N-TS) algorithm performs well relative to most other algorithms but is outperformed by GP-TS-M.

Next, we examine the symmetric and left-skewed distributions, Beta(2,2) and Beta(9,2), respectively. While the ordering of algorithms' performance is similar to the right-skewed Beta(2,9) case, we observe that the differences between algorithms begin to shrink. This

[29] There are two sources of variation between simulations: one from the algorithm itself and another from the exact WTP draws from the true distribution. Using expected rewards from playing a price reduces the variation caused by WTP draws, allowing for a cleaner comparison of algorithmic performance. Misra et al. (2019) also use expected rewards in their analysis.

Figure 4 **Cumulative Percent of Optimal Rewards (Profits)**



a) 5 Arms

b) 10 Arms

c) 100 Arms

Notes. The lines represent the means of the cumulative expected percentage of optimal rewards across 1000 simulations. The black horizontal line represents the maximum obtainable reward given the price set, while 100% represents the true optimal reward given the underlying distribution.

Table 3 Cumulative Percent of Optimal Rewards (Profits)

	After 500 Consumers							
	% of Price Set Maximum Reward				% of True Optimal Reward			
Algorithm	**B(2,9)**	**B(2,2)**	**B(9,2)**	**Mean**	**B(2,9)**	**B(2,2)**	**B(9,2)**	**Mean**
	5 Arms							
TS	72.7	91.2	97.6	**87.2**	65.6	87.7	97.2	**83.5**
GP-TS	66.7	92.0	95.3	**84.7**	60.2	88.4	94.9	**81.2**
GP-TS-M	87.3	94.6	96.3	**92.7**	78.7	90.9	95.9	**88.5**
N-TS	82.0	73.3	49.7	**68.3**	73.9	70.5	49.5	**64.6**
UCB	27.0	90.0	96.5	**71.1**	24.3	86.5	96.1	**69.0**
GP-UCB	26.6	92.0	96.6	**71.8**	24.0	88.5	96.2	**69.6**
GP-UCB-M	71.1	93.1	97.2	**87.1**	64.1	89.5	96.8	**83.5**
	10 Arms							
TS	51.0	82.8	93.6	**75.8**	46.9	82.5	93.2	**74.2**
GP-TS	61.1	90.4	94.3	**81.9**	56.2	90.1	93.9	**80.1**
GP-TS-M	90.3	93.5	95.3	**93.0**	83.1	93.3	94.9	**90.4**
N-TS	43.5	76.8	85.6	**68.6**	40.0	76.5	85.2	**67.3**
UCB	14.6	79.8	91.4	**61.9**	13.5	79.6	91.0	**61.3**
GP-UCB	27.0	89.8	95.6	**70.8**	24.8	89.5	95.2	**69.8**
GP-UCB-M	79.1	90.9	96.1	**88.7**	72.9	90.7	95.7	**86.4**
	100 Arms							
TS	1.0	43.5	69.0	**37.8**	1.0	43.5	69.0	**37.8**
GP-TS	60.6	90.3	94.3	**81.8**	60.6	90.3	94.3	**81.8**
GP-TS-M	87.4	93.3	95.3	**92.0**	87.4	93.3	95.3	**92.0**
N-TS	28.4	59.3	60.4	**49.3**	28.4	59.3	60.4	**49.3**
UCB	0.7	44.8	71.4	**39.0**	0.7	44.8	71.4	**39.0**
GP-UCB	20.5	88.1	94.8	**67.8**	20.5	88.1	94.8	**67.8**
GP-UCB-M	71.5	89.8	95.2	**85.5**	71.5	89.8	95.2	**85.5**
	After 2500 Consumers							
	% of Price Set Maximum Reward				**% of True Optimal Reward**			
Algorithm	**B(2,9)**	**B(2,2)**	**B(9,2)**	**Mean**	**B(2,9)**	**B(2,2)**	**B(9,2)**	**Mean**
	5 Arms							
TS	92.1	96.6	99.4	**96.1**	83.0	92.9	99.0	**91.7**
GP-TS	86.2	96.9	98.9	**94.0**	77.7	93.2	98.5	**89.8**
GP-TS-M	95.6	97.8	99.1	**97.5**	86.1	94.1	98.7	**93.0**
N-TS	94.6	80.3	50.2	**75.0**	85.3	77.2	50.0	**70.8**
UCB	75.4	96.6	99.1	**90.4**	68.0	92.9	98.7	**86.5**
GP-UCB	77.0	97.5	99.2	**91.2**	69.4	93.8	98.8	**87.3**
GP-UCB-M	90.8	98.0	99.3	**96.0**	81.9	94.2	98.9	**91.7**
	10 Arms							
TS	83.7	93.5	98.1	**91.8**	77.1	93.2	97.7	**89.3**
GP-TS	82.3	96.6	98.2	**92.4**	75.8	96.3	97.7	**89.9**
GP-TS-M	96.6	97.2	98.4	**97.4**	88.9	96.9	98.0	**94.6**
N-TS	76.5	90.7	95.6	**87.6**	70.4	90.4	95.2	**85.3**
UCB	52.4	91.9	97.3	**80.5**	48.3	91.6	96.9	**78.9**
GP-UCB	73.4	96.5	98.6	**89.5**	67.6	96.2	98.2	**87.3**
GP-UCB-M	93.5	96.8	98.7	**96.3**	86.1	96.5	98.3	**93.6**
	100 Arms							
TS	22.8	73.3	90.2	**62.1**	22.8	73.3	90.2	**62.1**
GP-TS	82.0	96.3	98.2	**92.2**	82.0	96.3	98.2	**92.2**
GP-TS-M	94.2	97.1	98.4	**96.6**	94.2	97.1	98.4	**96.6**
N-TS	32.2	67.4	74.6	**58.1**	32.2	67.4	74.6	**58.1**
UCB	12.4	70.8	87.2	**56.8**	12.4	70.8	87.2	**56.8**
GP-UCB	69.7	95.7	98.1	**87.8**	69.7	95.7	98.1	**87.8**
GP-UCB-M	90.1	96.0	98.0	**94.7**	90.1	96.0	98.0	**94.7**

finding aligns with expectations outlined in Section 3, as the learning problem becomes easier when the optimal price is higher within the price set, allowing all algorithms to perform well. The one exception is N-TS, which performs worse in these cases. Averaging across all three distributions, the best-performing algorithm is GP-TS-M, achieving 92.7%

of the maximum reward obtainable given the price set (and 88.5% of the true optimal) after 500 consumers and 97.5% (93.0% of the true optimal) after 2500 consumers (see Table 3).

We now increase the number of arms or prices. As the number of arms increases from 5 to 10, the rank ordering of the algorithms remains similar, with a few notable differences. In the Beta(2,9) case, the first informational externality allows GP-TS to initially outperform TS, though TS narrowly surpasses it by 2500 consumers. In contrast, GP-UCB now significantly outperforms UCB. Similarly, in both the Beta(2,2) and Beta(9,2) cases, the performance gap between GP-TS (GP-UCB) and TS (UCB) widens. Accordingly, the variation in performance across algorithms increases, with the lower-performing algorithms falling further behind the best-performing ones that use a GP. This growing difference occurs because of the first informational externality. Meanwhile, the second informational externality again provides a substantial gain in the Beta(2,9) case but only minor improvements in the Beta(2,2) and Beta(9,2) cases. GP-TS-M continues to lead in performance, while N-TS performs comparably to TS and UCB.

When the number of arms increases from 10 to 100, the learning problem becomes more challenging, and the above patterns intensify. The gap between the best and worst performing algorithms widens further, driven primarily by the benefits of the first informational externality, which grows as the number of arms increase. Learning across 100 arms without leveraging nearby information requires significantly more data, making the advantages of shared learning through a Gaussian process increasingly pronounced. The second informational externality maintains similar effects regardless of the number of arms, reinforcing the substantial gain in the Beta(2,9) case while providing only minor improvements in the Beta(2,2) and Beta(9,2) cases. GP-TS-M remains the best-performing algorithm, with N-TS still performing similarly to TS and UCB.

Having discussed the results across price granularities (number of arms), a question remains: in practice, a priori, how should a firm choose the price set? The general trade-off is that learning the optimal price is easier (more efficient) with fewer prices, but a higher optimum is possible when more prices are considered.

To illustrate, in the Beta(9,2) case with 5 arms and after 2500 consumers, GP-TS-M achieves 99.1% of the maximum reward possible given the chosen price set, which equates to 98.7% of the true optimal reward. Meanwhile, at the higher price granularity of 100 arms, GP-TS-M achieves 98.4% of both the maximum reward possible and the true optimal reward. In this case, the maximum reward possible with 5 arms happens to be close enough to the true optimal that the benefits of an easier learning problem outweigh the potential gains from being able to select a price closer to the true optimal. This is not always true. In the Beta(2,9) case, the maximum reward possible is just 90.2% of the true optimal. As a result, despite GP-TS-M obtaining 95.6% of the maximum reward possible with 5 arms, it achieves just 86.1% of the true optimal – much worse than 100 arms which achieves 94.2%. Thus, one important benefit of our proposed approach is that it makes it feasible to have a larger number of arms, enabling more precise discovery of the optimal price.

Overall, averaged across the three distributions, the highest-performing algorithm relative to the true optimal is GP-TS-M with 100 arms. Thus, we suggest firms use many arms, as this allows for higher potential gains that outweigh the losses from a more challenging learning problem, which are significantly mitigated by the incorporation of informational externalities.[30] Additionally, GP-TS-M with 100 arms is also the best-performing algorithm after 500 consumers, making this advice applicable for experiments of wide range of reasonable durations.[31]

6.1.2. Uplifts: Performance Increase from Informational Externalities We next examine the uplifts (the percentage improvement in cumulative profits from including an informational externality) in Table 4. This analysis quantitatively assesses the value of the externalities and clarifies how their impact depends on valuation distributions and the numbers of arms. Overall, we observe that while the uplift from the first externality is most influenced by the number of arms, the uplift from the second externality is primarily affected by the underlying WTP distribution.

After 2500 consumers, the benefit of incorporating the first informational externality using GPs into both TS and UCB increases with the number of arms across all three distributions. Averaged across the distributions, the uplift for adding GP to TS is -2.1% for 5 arms, 0.6% for 10 arms, and 48.4% for 100 arms. For UCB, the corresponding uplifts are 1.0%, 11.1%, and 54.7%. For 100 arms, a common pattern for both TS and UCB emerges: the uplift sharply decreases as the optimal price increases, shifting from the

[30] Across the three distributions we tested, with 100 arms the maximum reward obtainable was at least 99.99% of the true optimal, suggesting that 100 arms may be sufficient. Further testing on a wider variety of WTP distributions would be useful.

[31] If the experiment becomes sufficiently short, there will eventually be a point where a smaller price set will perform better.

Table 4 Uplifts in Performance from Informational Externalities

				After 500 Consumers			
			TS			UCB	
		5 Arms	10 Arms	100 Arms	5 Arms	10 Arms	100 Arms
Uplift from 1st externality (GP compared to base algos)	**B(2,9)**	-8.0% (-8.6, -7.3)	20.6% (19.2, 22.1)	5940% (5840, 6040)	2.0% (0.0, 4.1)	92.1% (86.6, 97.6)	2720% (2650, 2790)
	B(2,2)	1.0% (0.6, 1.3)	9.4% (9.0, 9.8)	108% (107, 109)	2.4% (2.0, 2.7)	12.6% (12.1, 13.1)	96.6% (96.0, 97.2)
	B(9,2)	-2.4% (-2.5, -2.2)	0.8% (0.6, 1.0)	36.8% (36.6, 37.1)	0.1% (0.0, 0.3)	4.6% (4.5, 4.8)	32.7% (32.6, 32.8)
	Mean	**-2.8% (-3.1, -2.6)**	**8.2% (7.8, 8.5)**	**116% (115, 117)**	**0.9% (0.7, 1.2)**	**14.3% (13.9, 14.7)**	**73.9% (73.4, 74.4)**
Uplift from 2nd externality (GP-M compared to GP algos)	**B(2,9)**	31.5% (30.5, 32.5)	50.1% (48.4, 51.8)	45.5% (44.1, 47.0)	176% (170, 182)	237% (222, 252)	296% (281, 310)
	B(2,2)	2.9% (2.6, 3.2)	3.5% (3.2, 3.9)	3.4% (3.1, 3.8)	1.3% (1.0, 1.6)	1.4% (1.0, 1.8)	2.1% (1.6, 2.5)
	B(9,2)	1.0% (0.9, 1.2)	1.1% (1.0, 1.2)	1.1% (0.9, 1.2)	0.6% (0.5, 0.7)	0.5% (0.4, 0.6)	0.5% (0.4, 0.5)
	Mean	**9.6% (9.3, 9.8)**	**13.7% (13.3, 14.0)**	**12.6% (12.3, 12.9)**	**21.5% (21.1, 21.9)**	**25.6% (25.1, 26.0)**	**26.3% (25.9, 26.7)**
Uplift from both externalities (GP-M compared to base algos)	**B(2,9)**	20.4% (19.6, 21.3)	78.5% (76.9, 80.0)	8610% (8470, 8740)	174% (168, 179)	468% (457, 479)	975% (970, 980)
	B(2,2)	3.8% (3.5,4.1)	13.2% (12.7, 13.6)	115% (114,116)	3.6% (3.2, 3.9)	14.1% (13.6, 14.5)	100% (100, 101)
	B(9,2)	-1.4% (-1.5, -1.2)	1.9% (1.7, 2.1)	38.3% (38.0, 38.6)	0.8% (0.7, 0.9)	5.2% (5.0, 5.3)	33.3% (33.2, 33.4)
	Mean	**6.4% (6.1, 6.6)**	**22.9% (22.5, 23.2)**	**143% (143, 144)**	**22.6% (22.1, 23.0)**	**43.3% (42.9, 43.7)**	**119% (119, 120)**
				After 2500 Consumers			
			TS			**UCB**	
		5 Arms	10 Arms	100 Arms	5 Arms	10 Arms	100 Arms
Uplift from 1st externality (GP compared to base algos)	**B(2,9)**	-6.4% (-6.5, -6.2)	-1.6% (-1.9, -1.3)	262% (260, 264)	2.2% (1.9, 2.4)	40.4% (39.7, 41.1)	464% (462, 467)
	B(2,2)	0.3% (0.1, 0.4)	3.4% (3.2, 3.5)	31.4% (31.2, 31.6)	1.0% (0.9, 1.1)	5.0% (4.8, 5.1)	35.2% (35.0, 35.4)
	B(9,2)	-0.5% (-0.6, -0.5)	0.0% (0.0, 0.1)	8.8% (8.7, 8.9)	0.1% (0.0, 0.1)	1.3% (1.3, 1.4)	12.5% (12.4, 12.6)
	Mean	**-2.1% (-2.2, -2.0)**	**0.6% (0.5, 0.8)**	**48.4% (48.2, 48.6)**	**1.0% (0.9, 1.1)**	**11.1% (11.0, 11.3)**	**54.7% (54.5, 54.9)**
Uplift from 2nd externality (GP-M compared to GP algos)	**B(2,9)**	10.9% (10.6, 11.1)	17.4% (17.1, 17.8)	14.9% (14.6, 15.2)	18.0% (17.6, 18.5)	27.6% (27.1, 28.1)	29.4% (28.9, 30.0)
	B(2,2)	1.0% (0.9, 1.1)	0.7% (0.5, 0.8)	0.9% (0.7, 1.0)	0.4% (0.3, 0.5)	0.3% (0.2, 0.5)	0.3% (0.1, 0.4)
	B(9,2)	0.2% (0.2, 0.3)	0.3% (0.2, 0.3)	0.3% (0.2, 0.3)	0.1% (0.1, 0.2)	0.2% (0.1, 0.2)	-0.1% (-0.1, 0.0)
	Mean	**3.7% (3.6, 3.8)**	**5.5% (5.4, 5.6)**	**4.8% (4.7, 4.9)**	**5.3% (5.1, 5.4)**	**7.7% (7.5, 7.8)**	**7.8% (7.7, 7.9)**
Uplift from both externalities (GP-M compared to base algos)	**B(2,9)**	3.8% (3.6, 4.0)	15.4% (15.2, 15.6)	315% (313, 318)	20.5% (20.0, 21.0)	78.9% (78.1, 79.7)	629% (627, 632)
	B(2,2)	1.3% (1.1, 1.4)	4.0% (3.9, 4.2)	32.5% (32.3, 32.7)	1.4% (1.3, 1.6)	5.3% (5.1, 5.5)	35.6% (35.4, 35.8)
	B(9,2)	-0.3% (-0.4, -0.3)	0.3% (0.2, 0.4)	9.0% (8.9, 9.1)	0.2% (0.2, 0.3)	1.5% (1.4, 1.5)	12.4% (12.3, 12.5)
	Mean	**1.5% (1.4, 1.6)**	**6.2% (6.1, 6.3)**	**55.5% (55.4, 55.7)**	**6.3% (6.2, 6.4)**	**19.6% (19.5, 19.8)**	**66.8% (66.6, 66.9)**

Notes. The table provides mean uplifts (averaged across 1000 simulations) along with their corresponding 99% confidence intervals, calculated using a paired t-test. The means are weighted.

right-skewed Beta(2,9) to the left-skewed Beta(9,2). These differences are substantial, with uplifts exceeding 200% for Beta(2,9) but dropping to just 9%–13% for Beta(9,2). This pattern persists for UCB across all numbers of arms, whereas for TS, the first externality actually decreases profits in the Beta(2,9) case for both 5 and 10 arms.

For the second informational externality, the uplift patterns are consistent within the TS and UCB families of algorithms, specifically for GP-TS to GP-TS-M and GP-UCB to GP-UCB-M. Across the various numbers of arms, the uplifts remain stable, ranging from 3.7% to 5.5% for the TS family and 5.3% to 7.8% for the UCB family. However, the uplifts vary substantially between distributions, with high gains concentrated in the Beta(2,9) case; they drop to under 1% for Beta(2,2) and approach 0% for Beta(9,2).

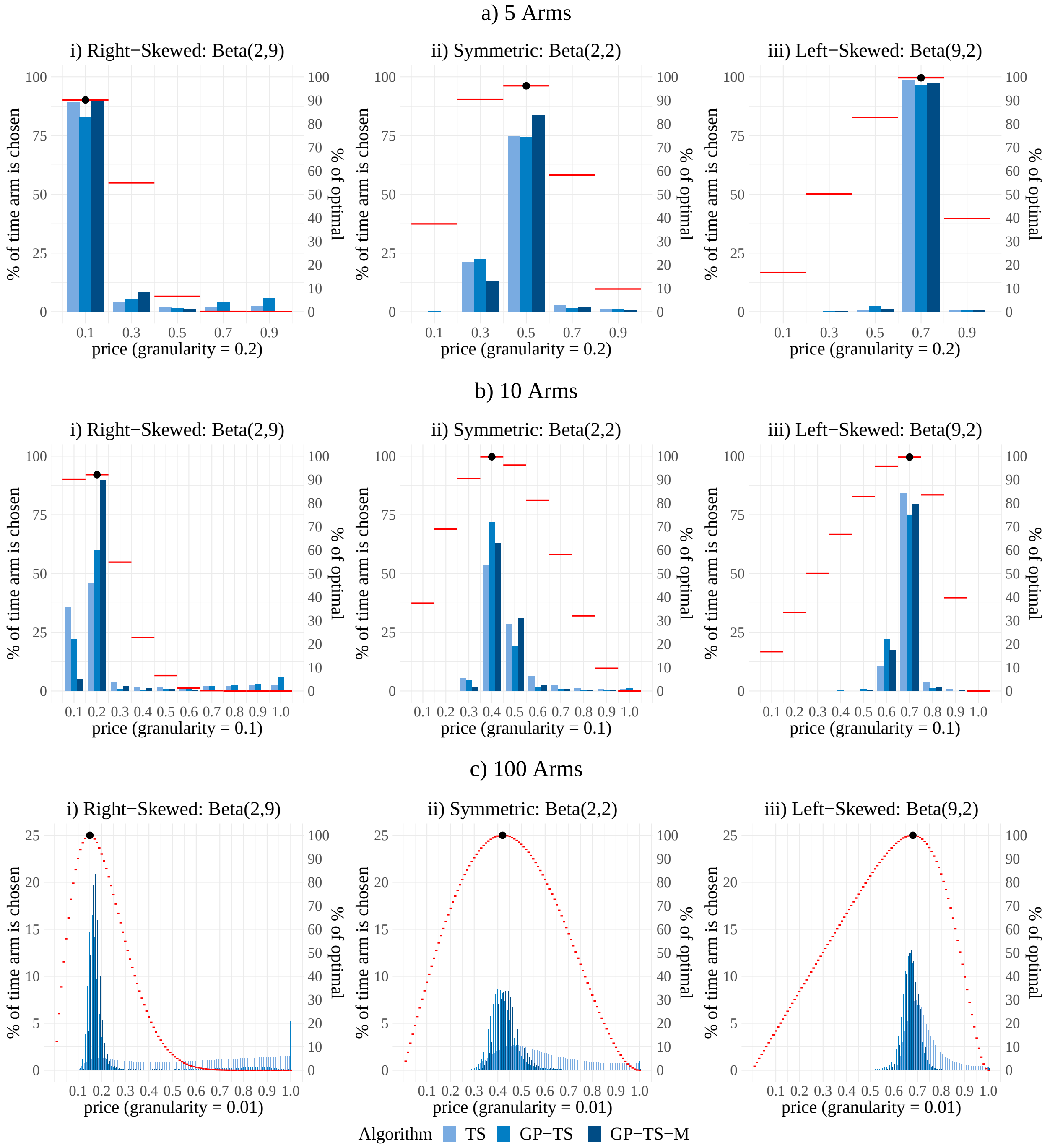
For 500 consumers, the uplift patterns are similar to those disccused earlier, but more pronounced. This is because comparative gains from algorithms often occur in the early rounds; over time, all algorithms eventually converge to the optimal price. This also implies that informational externalities are broadly valuable, regardless of whether the manager operates with a low experimentation budget (500 consumers) or a relatively high one (2500 consumers).

6.1.3. Mechanism: Investigating the Prices Chosen by Algorithms To understand why algorithms vary in performance, we examine the set of prices (arms) selected by each algorithm. Figure 5 presents three sets of three panels. Each graph includes both a left and a right y-axis. The left y-axis corresponds to the blue bar plots, which show the percentage of time each price was chosen. The right y-axis corresponds to the red horizontal lines, which represent the proportion of optimal rewards obtainable by selecting that price. The highest red line is further marked with a black dot, indicating the price that achieved the maximum reward within the price set as well as its percentage of the true optimal.

The value of the first informational externality (using a GP) increases with the number of arms, with the largest gains observed when there are 100 arms, while for 5 arms, the gains are negligible or even negative. This mechanism becomes clear when comparing the top row (5 arms) with the bottom row (100 arms) of Figure 5. Since TS does not account for dependence across arms, it must learn the reward for each price individually, requiring extensive testing of each price. In contrast, GP-TS pools information across many low-performing arms, enabling it to focus on higher-reward areas more quickly without repeatedly testing each price. As the number of arms increases, these advantages become substantial.

However, with just 5 arms, TS can learn the reward distribution for each price relatively quickly and may even outperform GP-TS. For example, with 5 arms and a Beta(2,9) distribution, GP-TS selects prices 0.7 and 0.9 (which provide minimal rewards) far more

Figure 5 **Histogram of Prices Played**



a) 5 Arms

b) 10 Arms

c) 100 Arms

Notes. Each subfigure has two y-axes, with price as the common x-axis. The left y-axis corresponds to the bar plots and represents the % of time each arm is chosen (based on 2500 consumers, averaged over 1000 simulations). The right y-axis corresponds to the red horizontal lines, which indicate the % of the true optimal reward obtainable at each corresponding price. The black dot marks the price that is optimal within the price set.

frequently than TS. When the arms or prices are far apart, the value of learning across arms can be negative. This is also exacerbated since the GP uses a single noise hyperparameter across all prices, leading to larger noise estimates in this price range. Consequently, the

GP requires more learning to eliminate these prices from consideration than TS. Appendix EC.6 outlines a method to address this issue by incorporating heterogeneity in noise for GP modeling.

Meanwhile, for the second informational externality, the largest gains occurred in the Beta(2,9) case, with positive uplifts diminishing as we move to the Beta(2,2) and Beta(9,2) cases. In the top-left panel (5 arms, right-skewed), all algorithms play the optimal price of 0.1 the majority of the time. However, both TS and GP-TS play prices 0.7 and 0.9 with a non-negligible frequency, whereas GP-TS-M almost never selects these prices. Since these arms provide nearly zero reward, playing them is quite costly, which highlights the superior performance of GP-TS-M.

To understand why this occurs, recall that profit is demand scaled by price, meaning that even if demands are learned equally across prices, uncertainty bounds around profits are increasing with price. Algorithms that do not consider monotonicity (such as TS and GP-TS) must invest significant resources to determine whether high prices are suboptimal. As monotonicity requires each demand curve to be monotonically decreasing, it eliminates many possible demand curves that TS and GP-TS cannot disregard. Specifically, it can lead to gains by excluding from consideration demand curves that upward-slope at high prices when the true optimal is a low price. Thus, with minimal data, monotonicity effectively reduces the need to explore high-noise, low-profit regions, thus demonstrating its value.

This advantage, however, applies primarily when the optimal price is low. Because lower prices have smaller reward bounds, other algorithms can also quickly dismiss low-profit, low-price arms; for example, the left-skewed panels of Figure 5 show that all of the algorithms rarely played prices below 0.5. Ultimately, the advantage from incorporating the second informational externality diminishes as the optimal price within the price set increases, although it remains positive across all simulations we ran.

6.1.4. Robustness: Alternative WTP Distributions We also conduct several analyses to assess the robustness of the method developed here. First, in Appendix EC.5.1, we test our method in an empirical setting using a demand curve estimated from field data. In this setting, the optimal price was low within the price set considered, and the results accordingly aligned with the Beta(2,9) case for 5, 10, and 100 arms. This lends credence to our main analysis and demonstrates the practical value and applicability of the method.

Next, since the GP-based method requires continuity of the demand curve, we consider cases where the demand curve has discontinuities at known prices – specifically, left-digit bias (Thomas and Morwitz 2005), where consumers perceive a larger price increase when the left-digit changes (e.g., from \$1.99 to \$2.00 is perceived as greater than a one-cent increase). We find that even in this challenging situation, the informational externalities improve performance (see Appendix EC.5.2). However, if the left-digit bias is large enough, it may be more effective for a firm to use only the prices immediately preceding the discontinuities as the price set rather than adopting a finer price granularity.

Finally, to examine whether the proposed bandit algorithms provide a long-run or persistent advantage, we explore the case of time-varying (seasonal) demand in Appendix EC.5.3, where we introduce an unknown demand shock each period. Once again, we find that informational externalities, especially monotonicity, significantly enhance long-term profits. Notably, we find that learning is so efficient that, when the shocks are large enough, GP-TS-M can achieve higher profits by restarting the bandit experiment at each shock rather than relying on prior data. For the other algorithms tested, however, the start-up cost outweighed the gains from more accurately learning the demand curve.

7. Conclusion and Future Research

We have proposed a method for efficient and robust learning of the relevant portion of the demand curve by using reinforcement learning in multi-armed bandits informed by microeconomic principles, specifically monotonicity of demand curves..

Notably, our algorithm outperformed baseline methods across a variety of scenarios, including number of arms and valuations distributions as well as challenging cases like discontinuous and time-varying demand, and achieved significant gains across both the short and long term horizons. Our method is particularly beneficial for managers with limited time for experimentation. By reducing the required experimentation time, our approach makes price experimentation more feasible, minimizing potential monetary losses and limiting negative consumer impact. Our method leverages informational externalities to enable testing a finer grid of prices, allowing firms to evaluate prices closer to the true optimal. This not only increases gains during the experiment but can also lead to long-term profit improvements when a near-optimal price is adopted after the experiment concludes.

There are several avenues to extend this research. First, the method could be adapted to settings involving multiple units purchased or multiple products, where consumer choices for one product inform the valuation distribution of related products. Second, applying the method to competitive environments, where competitors are experimenting with price levels and demand responses, would be valuable. More broadly, efficiently learning unknown demand curves with minimal experimental impact remains a significant challenge across many markets. This research contributes to addressing this challenge by closely integrating theory with algorithm development for effective learning.

Funding and Competing Interests

All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. The authors have no specific funding to report.

References

Aghion P, Bolton P, Harris C, Jullien B (1991) Optimal learning by experimentation. *The review of economic studies* 58(4):621–654.

Agrawal R (1995) Sample mean based index policies by o (log n) regret for the multi-armed bandit problem. *Advances in Applied Probability* 27(4):1054–1078.

Agrawal S, Goyal N (2012) Analysis of thompson sampling for the multi-armed bandit problem. *Conference on learning theory*, 39–1.

Aparicio D, Simester D (2022) Price frictions and the success of new products. *Marketing Science* 41(6):1057–1073.

Auer P (2002) Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research* 3(Nov):397–422.

Auer P, Cesa-Bianchi N, Fischer P (2002) Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47(2-3):235–256.

Bayati M, Hamidi N, Johari R, Khosravi K (2020) Unreasonable effectiveness of greedy algorithms in multi-armed bandit with many arms. *Advances in Neural Information Processing Systems* 33:1713–1723.

Bergemann D, Schlag KH (2008) Pricing without priors. *Journal of the European Economic Association* 6(2-3):560–569.

Besbes O, Zeevi A (2009) Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research* 57(6):1407–1420.

Botev Z, Belzile L (2021) *TruncatedNormal: Truncated Multivariate Normal and Student Distributions*. URL `https://CRAN.R-project.org/package=TruncatedNormal`, r package version 2.2.2.

Chapelle O, Li L (2011) An empirical evaluation of thompson sampling. *Advances in neural information processing systems*, 2249–2257.

Chatterjee S, Sen S (2021) Regret minimization in isotonic, heavy-tailed contextual bandits via adaptive confidence bands. URL `https://arxiv.org/abs/2110.10245`.

Chen Q, Jasin S, Duenyas I (2019) Nonparametric self-adjusting control for joint learning and optimization of multiproduct pricing with finite resource capacity. *Mathematics of Operations Research* 44(2):601–631.

Cheshire J, Ménard P, Carpentier A (2020) The influence of shape constraints on the thresholding bandit problem. *Conference on Learning Theory*, 1228–1275 (PMLR).

Ching AT, Osborne M (2020) Identification and estimation of forward-looking behavior: The case of consumer stockpiling. *Marketing Science* 39(4):707–726.

Chou C, Kumar V (2024) Estimating demand for subscription products: Identification of willingness to pay without price variation. *Marketing Science* .

Chowdhury SR, Gopalan A (2017) On kernelized multi-armed bandits. *International Conference on Machine Learning*, 844–853 (PMLR).

Cohen SN, Treetanthiploet T (2020) Asymptotic randomised control with applications to bandits. *arXiv preprint arXiv:2010.07252* .

Dann C, Mansour Y, Mohri M, Sekhari A, Sridharan K (2022) Guarantees for epsilon-greedy reinforcement learning with function approximation. *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 4666–4689 (PMLR).

Dholakia U (2015) The risks of changing your prices too often. *Harvard Business Review* .

Duvenaud D (2014) *Automatic model construction with Gaussian processes*. Ph.D. thesis.

Erdem T, Keane MP (1996) Decision-making under uncertainty: Capturing dynamic brand choice processes in turbulent consumer goods markets. *Marketing science* 15(1):1–20.

Ferreira KJ, Simchi-Levi D, Wang H (2018) Online network revenue management using thompson sampling. *Operations research* 66(6):1586–1602.

Filippi S, Cappe O, Garivier A, Szepesvári C (2010) Parametric bandits: The generalized linear case. *Advances in neural information processing systems* 23.

Furman J, Simcoe T (2015) The economics of big data and differential pricing. *The White House President Barack Obama* .

Gittins J (1974) A dynamic allocation index for the sequential design of experiments. *Progress in statistics* 241–266.

Goldberg PW, Williams CK, Bishop CM (1997) Regression with input-dependent noise: A gaussian process treatment. *Advances in neural information processing systems* 10:493–499.

Gordon BR, Zettelmeyer F, Bhargava N, Chapsky D (2019) A comparison of approaches to advertising measurement: Evidence from big field experiments at facebook. *Marketing Science* 38(2):193–225.

Guntuboyina A, Sen B (2018) Nonparametric shape-restricted regression. *Statistical Science* 33(4):568–594.

Handel BR, Misra K (2015) Robust new product pricing. *Marketing Science* 34(6):864–881.

Hanssens DM, Pauwels KH (2016) Demonstrating the value of marketing. *Journal of marketing* 80(6):173–190.

Hauser JR, Urban GL, Liberali G, Braun M (2009) Website morphing. *Marketing Science* 28(2):202–223.

Hendel I, Nevo A (2006) Measuring the implications of sales and consumer inventory behavior. *Econometrica* 74(6):1637–1673.

Hill DN, Nassif H, Liu Y, Iyer A, Vishwanathan S (2017) An efficient bandit algorithm for realtime multivariate optimization. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1813–1821.

Hoban PR, Bucklin RE (2015) Effects of internet display advertising in the purchase funnel: Model-based insights from a randomized field experiment. *Journal of Marketing Research* 52(3):375–393.

Hoffman M, Brochu E, De Freitas N, et al. (2011) Portfolio allocation for bayesian optimization. *UAI*, 327–336 (Citeseer).

Huang J, Reiley D, Riabov N (2018) Measuring consumer sensitivity to audio advertising: A field experiment on pandora internet radio. *Available at SSRN 3166676* .

Huang Y, Ellickson PB, Lovett MJ (2022) Learning to set prices. *Journal of Marketing Research* 59(2):411–434.

Jindal P, Zhu T, Chintagunta P, Dhar S (2020) Marketing-mix response across retail formats: the role of shopping trip types. *Journal of Marketing* 84(2):114–132.

Kawale J, Bui HH, Kveton B, Tran-Thanh L, Chawla S (2015) Efficient thompson sampling for online matrix-factorization recommendation. *Advances in neural information processing systems*, 1297–1305.

Kermisch R, Burns D (2018) Is pricing killing your profits. *Bain & Company Report, Bain: Boston* .

Kersting K, Plagemann C, Pfaff P, Burgard W (2007) Most likely heteroscedastic gaussian process regression. *Proceedings of the 24th international conference on Machine learning*, 393–400.

Lai TL, Robbins H (1985) Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics* 6(1):4–22.

Lambrecht A, Tucker C, Wiertz C (2018) Advertising to early trend propagators: Evidence from twitter. *Marketing Science* 37(2):177–199.

Maatouk H, Bay X (2017) Gaussian process emulators for computer experiments with inequality constraints. *Mathematical Geosciences* 49(5):557–582.

Miao S, Wang Y (2024) Demand balancing in primal-dual optimization for blind network revenue management. URL `https://arxiv.org/abs/2404.04467`.

Micchelli CA, Xu Y, Zhang H (2006) Universal kernels. *Journal of Machine Learning Research* 7(12).

Misra K, Schwartz EM, Abernethy J (2019) Dynamic online pricing with incomplete information using multiarmed bandit experiments. *Marketing Science* 38(2):226–252.

Murray I (2008) Introduction to gaussian processes. *Dept. Computer Science, University of Toronto* .

Nair H (2007) Intertemporal price discrimination with forward-looking consumers: Application to the us market for console video-games. *Quantitative Marketing and Economics* 5(3):239–292.

Oren SS, Smith SA, Wilson RB (1982) Nonlinear pricing in markets with interdependent demand. *Marketing Science* 1(3):287–313.

Rao RC, Bass FM (1985) Competition, strategy, and price dynamics: A theoretical and empirical investigation. *Journal of Marketing Research* 22(3):283–296.

Rebonato R, Jäckel P (2011) The most general methodology to create a valid correlation matrix for risk management and option pricing purposes. *Available at SSRN 1969689* .

Ringbeck D, Huchzermeier A (2019) Dynamic pricing and learning: An application of gaussian process regression. *Available at SSRN 3406293* .

Rothschild M (1974) A two-armed bandit theory of market pricing. *Journal of Economic Theory* 9(2):185–202.

Rubel O (2013) Stochastic competitive entries and dynamic pricing. *European Journal of Operational Research* 231(2):381–392.

Russo D, Van Roy B (2014) Learning to optimize via posterior sampling. *Mathematics of Operations Research* 39(4):1221–1243.

Sahni NS, Nair HS (2020) Does advertising serve as a signal? evidence from a field experiment in mobile search. *The Review of Economic Studies* 87(3):1529–1564.

Schwartz EM, Bradlow ET, Fader PS (2017) Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science* 36(4):500–522.

Shahriari B, Swersky K, Wang Z, Adams RP, De Freitas N (2015) Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE* 104(1):148–175.

Simester D, Hu Y, Brynjolfsson E, Anderson ET (2009) Dynamics of retail advertising: Evidence from a field experiment. *Economic Inquiry* 47(3):482–499.

Soysal GP, Krishnamurthi L (2012) Demand dynamics in the seasonal goods industry: An empirical analysis. *Marketing Science* 31(2):293–316.

Srinivas N, Krause A, Kakade SM, Seeger M (2009) Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995* .

Strulov-Shlain A (2023) More than a penny's worth: Left-digit bias and firm pricing. *Review of Economic Studies* 90(5):2612–2645.

Thomas M, Morwitz V (2005) Penny wise and pound foolish: the left-digit effect in price cognition. *Journal of Consumer Research* 32(1):54–64.

Thompson WR (1933) On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25(3/4):285–294.

Tirole J (1988) *The theory of industrial organization* (MIT press).

Urteaga I, Wiggins CH (2022) Nonparametric gaussian mixture models for the multi-armed bandit. *arXiv preprint arXiv:1808.02932* .

Wang Y, Chen B, Simchi-Levi D (2021) Multimodal dynamic pricing. *Management Science* 67(10):6136–6152.

Williams CK, Rasmussen CE (2006) *Gaussian processes for machine learning*, volume 2 (MIT press Cambridge, MA).

Yu M, Debo L, Kapuscinski R (2016) Strategic waiting for consumer-generated quality information: Dynamic pricing of new experience goods. *Management Science* 62(2):410–435.

Zhang L, Chung DJ (2020) Price bargaining and competition in online platforms: An empirical analysis of the daily deal market. *Marketing Science* 39(4):687–706.

Zhao H, He J, Zhou D, Zhang T, Gu Q (2023) Variance-dependent regret bounds for linear bandits and reinforcement learning: Adaptivity and computational efficiency. *The Thirty Sixth Annual Conference on Learning Theory*, 4977–5020 (PMLR).

Electronic Companion Supplement

EC.1 Gaussian Process Regression

This section provides an overview of Gaussian process regression. Formally, the assumption is that the demand function D is jointly Gaussian-distributed and completely defined by its mean $\mu(p)$ and covariance function $k(p,p')$, such that $D(p) \sim \mathrm{GP}(\mu(p), k(p,p'))$. The mean and covariance functions are defined as follows (Williams and Rasmussen 2006):

$$\mu(p) = \mathbb{E}[D(p)], \tag{EC.1}$$

$$k(p,p') = \mathbb{E}[(D(p)-\mu(p))(D(p')-\mu(p'))]. \tag{EC.2}$$

For ease of exposition, we set $\mu(p) = 0$.[32]

The kernel can then be used to compute a covariance matrix $K(P,P)$, which contains the covariance between all test points, as well as a covariance matrix (either $K(P,P_t)$ or $K(P_t,P)$) between training and test cases. The joint distribution of the training data P_t and the test points P can be written as follows (equation (2.21) in Williams and Rasmussen (2006)):

$$\begin{pmatrix} y_t \\ D^* \end{pmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(P_t,P_t)+\sigma_y^2 I & K(P_t,P) \\ K(P,P_t) & K(P,P) \end{bmatrix}\right), \tag{EC.3}$$

where $D^* = D(P)$ is a random variable denoting the GP predictions at test points P. It then follows from equations (2.22–2.24) in Williams and Rasmussen (2006) that:

$$D^* | P_t, y_t, P \sim N(\mu(D^*), \mathrm{Cov}(D^*)), \text{ where} \tag{EC.4}$$

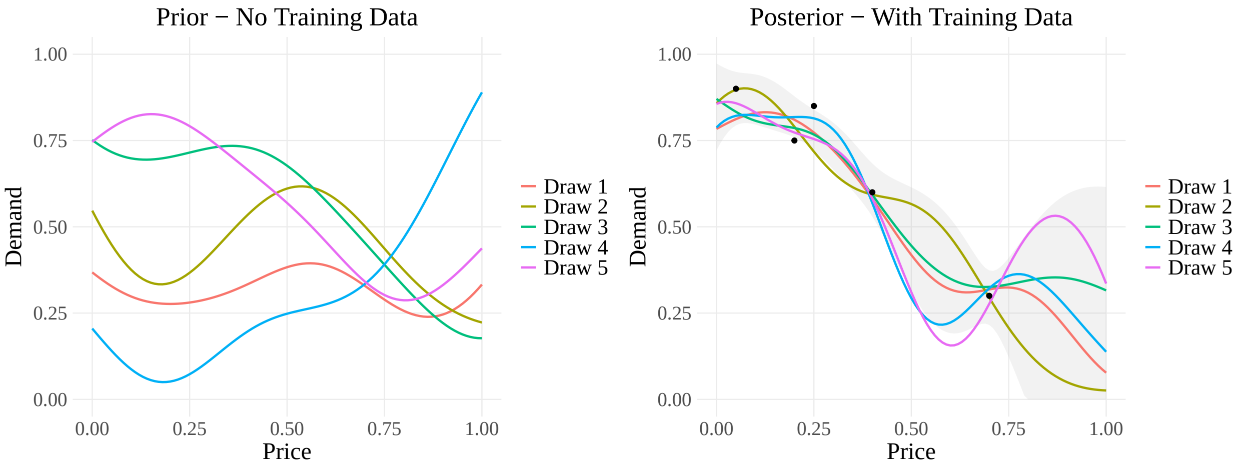
$$\mu(D^*) = K(P,P_t)[K(P_t,P_t)+\sigma_y^2 I]^{-1} y_t, \tag{EC.5}$$

$$\mathrm{Cov}(D^*) = K(P,P) - K(P,P_t)[K(P_t,P_t)+\sigma_y^2 I]^{-1} K(P_t,P). \tag{EC.6}$$

Figure EC.1 illustrates a simple example of GP regression. As data is obtained, the space where the true demand function could exist becomes more constrained. Accordingly, the range of uncertainty is smaller in areas closer to data points compared to those farther away, as shown by the shaded area representing the 95% confidence intervals at the test points.

[32] If we have a GP, $D \sim GP(\mu,k)$, where the prior mean function is non-zero, then $D' = D - \mu$ is the zero-mean Gaussian process, $D' \sim GP(0,k)$. Hence, using observations from the values of D, we can subtract the prior mean function values to obtain observations of D', perform inference on D', and then add back the prior mean $\mu(P)$ to the posterior mean to recover the posterior of D.

Figure EC.1 Random Samples from Gaussian Process With and Without Training Data



Notes. Lines represent five random draws from the GP in both the prior and posterior. In the prior, the mean was set to 0.5. For both the prior and posterior, the RBF kernel was used with hyperparameters $l = 0.2$, $\sigma_D^2 = 0.08$, and $\sigma_y^2 = 0.0016$. There were 101 test points, $P = \{0, 0.01, 0.02, \ldots, 1\}$. The five draws from the posterior distribution were sampled from the GP with training data $P_t = \{0.05, 0.2, 0.25, 0.4, 0.7\}$ and $y_t = \{0.9, 0.75, 0.85, 0.6, 0.3\}$. The shaded area represents the 95% confidence interval at each test point.

EC.2 Computational Issues

A computational issue that arises in fitting a posterior Gaussian process is that matrix inversion is $\mathcal{O}(n^3)$, implying it does not scale well to larger datasets. Typically, the training data grows with each purchase observation (thus depending on t), but we mitigate this issue because purchases can only be observed at prices within the fixed price set. This allows us to set the input training data as the set of test prices and the associated output training data as the observed purchase rates. The only additional adjustment needed is to the noise hyperparameter. As the sample variance scales with the number of observations, the noise hyperparameter is specified accordingly: $\sigma_y^2 = \{0.25/n_{1t}, \ldots, 0.25/n_{At}\}$. This approach ensures that computational complexity does not increase with the number of purchase observations but instead depends on the size of the initial set of test prices.

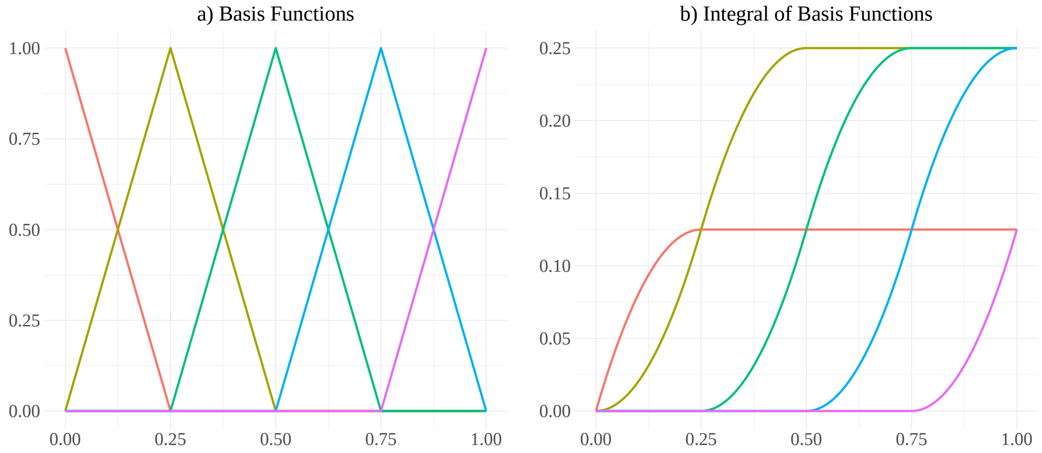
Additionally, one common issue when running GP algorithms is floating-point precision errors, which can lead to negative eigenvalues and violate the positive semi-definite property of covariance matrices. We follow the approach devised by Rebonato and Jäckel (2011) to obtain the nearest covariance matrix.

EC.3 Implementation of Monotonic GP Bandits

EC.3.1 Basis Function Visualization

Consider an example where $N = 4$ meaning there are 5 equally spaced knots $\{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$. Figure EC.2 shows the five basis functions along with their corresponding integrals.

Figure EC.2 Plot of Basis Functions and their Integrals (5 knots)



EC.3.2 Derivation of Proposition 1

Lemma EC.1. *The distance between a continuous function D and its estimate via basis functions $D_J(p) = \sum_{j=0}^{J} D(u_j)h_j(p)$ converges to 0 as $J \to \infty$.*

From Lemma EC.1, a continuous demand function D can be estimated via basis functions as $D(p) \approx \sum_{j=0}^{J} D(u_j)h_j(p)$. Since, by assumption, the derivative of D is also continuous, the same formula can be used to estimate D' as follows:

$$D'(p) \approx \sum_{j=0}^{J} D'(u_j)h_j(p). \tag{EC.7}$$

Additionally, by the Fundamental Theorem of Calculus,

$$D(p) - D(0) = \int_0^p D'(t)\,dt. \tag{EC.8}$$

Substituting (EC.7) into (EC.8) gives

$$D(p) \approx D(0) + \sum_{j=0}^{J} D'(u_j) \int_0^p h_j(t)\,dt. \quad \square \tag{EC.9}$$

EC.3.3 Gaussian Process – Estimation of Derivatives and Intercept

The basis function method requires the estimation of the intercept and the derivatives at each of the prices in the consideration set. More formally, the goal is to estimate the posterior mean and covariance for $\{D(0), D'(p_1), \ldots, D'(p_A)\}$. Temporarily ignoring the intercept, the key insight is that the derivatives of a GP are also a GP. This implies that D'^*, the posterior vector of derivatives of D^*, is given by:

$$D'^* \sim N\left(\frac{d}{dp}\mu(D^*), \frac{d}{dp}\,\mathrm{Cov}(D^*)\right). \tag{EC.10}$$

Since the values of D^* are only at the test points P, the derivative only needs to be calculated with respect to P. Consequently, to estimate the posterior mean and covariance for $\{D(0), D'(p_1), \ldots, D'(p_A)\}$, the only necessary adjustment is to compute the derivatives of the kernel function with respect to the test points.

We compute the partial derivatives of the kernel with respect to the prices as follows:

$$\frac{\partial k(p_i^*, p_j)}{\partial p_i^*} = \frac{\sigma_D^2}{l^2} (p_j - p_i^*) \exp\left(\frac{-(p_i^* - p_j)^2}{2l^2}\right), \tag{EC.11}$$

$$\frac{\partial k(p_i, p_j^*)}{\partial p_j^*} = \frac{\sigma_D^2}{l^2} (p_i - p_j^*) \exp\left(\frac{-(p_i - p_j^*)^2}{2l^2}\right), \tag{EC.12}$$

$$\frac{\partial^2 k(p_i^*, p_j^*)}{\partial p_i^* \partial p_j^*} = \frac{\sigma_D^2}{l^4} \left(l^2 - (p_i^* - p_j^*)^2\right) \exp\left(\frac{-(p_i^* - p_j^*)^2}{2l^2}\right). \tag{EC.13}$$

EC.4 Regret Bounds

Consider a setting where $P = \{p_1 \le p_2 \le \cdots \le p_d\}$ are a subset of prices that we choose as knots. We consider a Gaussian process for which draws are C^1 almost surely, and consider the joint distribution over draws f, f'. We define the set of monotonic functions,

$$\mathcal{M} = \{f \in C^1([0,1]) : f'(x) \le 0, x \in [0,1]\}.$$

Ideally we would like to restrict draws of our GP to $\mathcal{M}$, but in general, since we can only evaluate our GP at a finite set of points, we instead insist that our function is monotonic at the set of knots,

$$\mathcal{M}(P) = \{f(\in C^1[0,1] : f'(p) \le 0, p \in P\}$$

We denote the joint prior distribution over the function and the derivative $\Pi_0 = GP([0,0], K|\mathcal{M}(P))$, where K is the appropriate kernel.

Let $p^* = \arg\max_{p \in P} f(p)$ – note that since f is drawn from the underlying prior, p^* is a random variable. We define the Bayesian regret of our policy

$$BR_T = \sum_{t=1}^{T} \mathbb{E}[p^* f(p^*) - p_t f(p_t)]$$

where the expectation is over draws from the prior, reward noise, and any internal randomness of the algorithm.

Throughout the following, we refer to the truncated distribution as Π_t, and the untruncated distribution $GP([\mu_t, \mu_t'], K_t)$ as $\Pi_{t,u}$. Let the induced probability laws and expecation, with respect to the measure conditioned on the history $\mathcal{H}_t = \{(p_s, r_s)\}_{s=1}^t$, be $\mathbb{P}_t, \mathbb{E}_t$, and for the untruncated version, $\mathbb{P}_{t,u}, \mathbb{E}_{t,u}$. Critically we make the following assumption:

Assumption 1. The probability of returning a function monotonic on the knots is bounded below, i.e., there exists $c \geq 0$ such that

$$\mathbb{P}_{t,u}(f_t, f_t' \in \mathcal{M}(P)) \geq c, \forall t \geq 1$$

Remark: Note that $\mathbb{P}_{t,u}(f_t'(P) \geq 0)$ is equivalent to $\mathbb{P}(y \geq 0)$ for $x, y \sim N([\mu_t(P), \mu_t'(P)], K_t)$. This is an integral of a multivariate Gaussian over an open set. Since $f'(P) \geq 0$ by definition of the prior, if $\mu'(P) \to f'(P)$, we should expect this probability to actually increase with t.

Additionally, we require the following lemma linking a distribution with its truncated version.

Lemma EC.2. *Let $p(x)$ be a distribution on $\mathbb{R}^d$. Let $S \subset \mathbb{R}^d$. Define the truncated pdf on $\mathbb{R}^d$, $p_S(x) = \mathbf{1}\{x \in S\} p(x)/\mu(S)$ where $\mu(S) = \int_{x \in S} p(x)$. Then given an event $E \subset \mathbb{R}^d$,*

$$\mathbb{P}_p(\mathbf{1}\{E \cap S\}) \leq \mathbb{P}_{p_S}(E) \leq \frac{1}{\mu(S)} \mathbb{P}_p(E)$$

and given a function $f(x) : \mathbb{R}^d \to \mathbb{R}$

$$\mathbb{E}_{p_S}(f(x)) \leq \frac{1}{\mu(S)} \mathbb{E}_p(f(x))$$

Proof. The lower bound is immediate since $\mu(S) \leq 1$.

$$\begin{aligned}
\mathbb{P}_{p_S}(E) &= \int_{x \in \mathbb{R}^d} \mathbf{1}\{x \in E\} \mathbf{1}\{x \in S\} p(x)/\mu(S) \\
&= \frac{1}{\mu(S)} \int_{x \in \mathbb{R}^d} \mathbf{1}\{x \in E\} \mathbf{1}\{x \in S\} p(x) \\
&\leq \frac{1}{\mu(S)} \int_{x \in \mathbb{R}^d} \mathbf{1}\{x \in E\} p(x) \\
&\leq \frac{1}{\mu(S)} \mathbb{P}_p(E)
\end{aligned}$$

The result on expectations is immediate. □

Theorem EC.1. *The Bayes Regret of Algorithm 1 GP-TS-M is bounded by*

$$\begin{aligned}
BR_t &\leq \mathbb{E}\left[\sqrt{\sum_{t=1}^{\infty} p_t^2}\right] \sqrt{\gamma_T \log(1 + \sigma_0^{-2}) \log\left(\frac{T^2|A|}{\sqrt{2\pi}}\right)} + \mathcal{E}_T + 1 \\
&\leq \sqrt{T\gamma_T \log(1 + \sigma_0^{-2}) \log\left(\frac{T^2|A|}{\sqrt{2\pi}}\right)} + \mathcal{E}_t + 1
\end{aligned}$$

where $\mathcal{E}_T = \sum_{t=1} \mathbb{E}[\mu_t(p^*) - \mathbb{E}_t[f_t(p^*)]]$ and γ_T is the mutual information of the Gaussian process (Srinivas et al. 2009).

Interpreting the Regret: We remark that if we were not restricting to the monotonic set of functions, the argument shows that $\mathcal{E}_T$ is 0. And so the final regret is of the form $O(\sqrt{\gamma_T T \log(T|P|}))$. Note that this regret is independent of the underlying constraint set of monotonic functions. To understand the impact of the underlying constraint set, we focus on the path-dependent regret term $\sum_{t=1}^{T} p_t^2$. In general, this quantity is less than the maximum price played times T, and is a tighter regret result compared to existing works. We remark that the looseness in this result is primarily due to using loose tail bounds that do not effectively account for the constraint set. Future work could examine different and potentially tighter bounds.

Discussion of $\mathcal{E}_T$. In Figure EC.3, we have plotted $\log(t)$ vs. $\log(E_t)$, where E_t estimates each term of $\mathcal{E}_t$, defined as $\mathbb{E}[\mu_t(p^*) - \mathbb{E}_t[f_t(p^*)]]$. At each update of the Gaussian process, f_t is obtained from 10,000 posterior draws, and the difference from μ_t is calculated to estimate E_t. To ensure robustness, these E_t values were averaged across 1,000 independent simulations. As the plot demonstrates, $E_t \approx t^{-\alpha}$ where $\alpha \in [.5, .8]$. This implies that $\mathcal{E}_T \leq O(T^{-\alpha+1})$ hence contributing a regret bounded by $O(\sqrt{T})$. Intuitively this is not surprising, we should expect fast concentration of $\mu_t(p^*)$ and $\mathbb{E}_t[f_t(p^*)]$ to both quickly concentrate to $\mu(p^*)$ at a rate that matches the parametric rate of $1/\sqrt{t}$.

Proof. Define $U_t(p) := \mu_{t-1}(p) + \beta_{t-1}^{1/2}\sigma_{t-1}(p)$ where $\beta_t = \log(t^2 c^{-1}|P|/\sqrt{2\pi})$. Note that, conditioned on $\mathcal{H}_t$, the optimal action p^* and the action p_t selected by posterior sampling are identically distributed by Fact 5 (see below). In addition, U_t is deterministic conditioned on the history, so, $\mathbb{E}_t[U_t(p^*)] = \mathbb{E}_t[U_t(A_t)]$. Therefore,

$$\begin{aligned}
\mathbb{E}[p^* f(p^*) - p_t f(p_t)] &= \mathbb{E}[\mathbb{E}_t[p^* f(p^*) - p_t f(p_t)]] \\
&= \mathbb{E}[\mathbb{E}_t[p_t U_t(p_t) - p^* U_t(p^*) + p^* f(p^*) - p_t f(p_t)]] \\
&= \mathbb{E}[\mathbb{E}_t[p_t U_t(p_t) - p_t f(p_t)] + \mathbb{E}_t[p^* f(p^*) - p^* U_t(p^*)]] \\
&= \mathbb{E}[p_t U_t(p_t) - p_t f(p_t)] + \mathbb{E}[p^* f(p^*) - p^* U_t(p^*)].
\end{aligned}$$

Thus, we see that we can bound the Bayes-Regret as

$$BR(T) \leq \sum_{t=1}^{T} \mathbb{E}[p_t U_t(p_t) - p_t f(p_t)] + \sum_{t=1}^{T} \mathbb{E}[p^* f(p^*) - p^* U_t(p^*)] \tag{EC.14}$$

$$\tag{EC.15}$$

We now focus on the first term,

$$\begin{aligned}
p_t U_t(p_t) - p_t f(p_t) &= p_t U_t(p_t) - p_t \mu_t(p_t) + p_t \mu_t(p_t) - p_t f(p_t) \\
&= \mathbb{E}[p_t U_t(p_t) - p_t \mu_t(p_t)] + p_t \mu_t(p_t) - p_t f(p_t) \\
&\leq p_t \beta_t^{1/2} \sigma_t(p_t) + p_t \mu_t(p_t) - p_t f(p_t) \\
&\leq p_t \beta_t^{1/2} \sigma_t(p_t) + \mu_t(p_t) - f(p_t)
\end{aligned}$$

Next,

$$\sum_{t=1}^{T} p_t \beta_t^{1/2} \sigma_t(p_t) \leq \sqrt{\beta_T \sum_{t=1}^{\infty} p_t^2} \sqrt{\sum_{t=1}^{\infty} \sigma_t^2(p_t)} \qquad \text{(Cauchy-Schwartz)}$$

A standard argument (see Srinivas et al. (2009)) shows that

$$\sum_{t=1}^{\infty} \sigma_t^2(p_t) \leq \frac{\gamma_T}{\log(1+\sigma^{-2})}$$

Finally, we bound the second term of EC.14

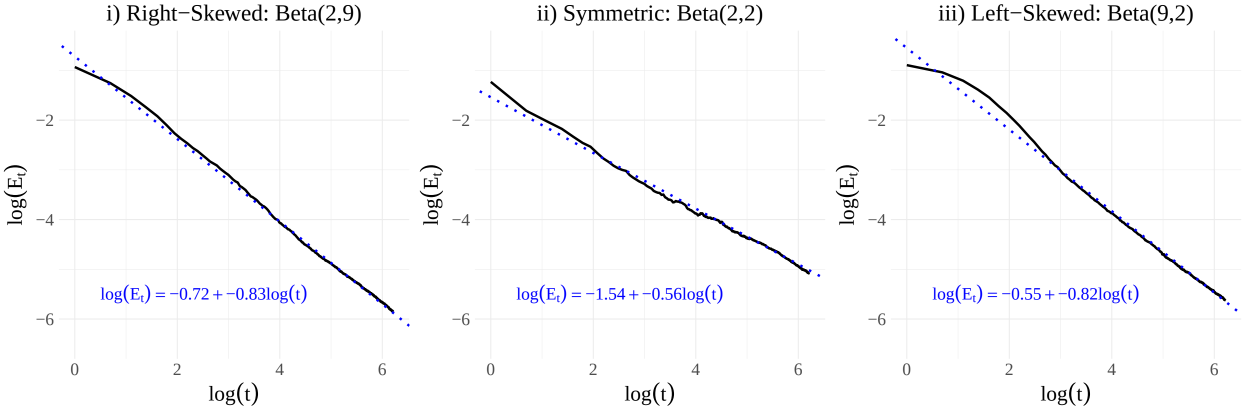
$$\begin{aligned}
\sum_{t=1}^{T} E[p^* f(p^*) - p^* U_t(p^*)] &\leq \sum_{t=1}^{\infty} \sum_{p \in P} \mathbb{E}_t[\mathbf{1}\{f(p) - U_t(p) \geq 0\}(f(p) - U_t(p))] \\
&\leq \sum_{t=1}^{\infty} \sum_{p \in P} \frac{1}{\mathbb{P}_{t,u}(M)} \mathbb{E}_{t,u}[\mathbf{1}\{f(p) - U_t(p) \geq 0\}(f(p) - U_t(p))] \\
&\leq \sum_{t=1}^{\infty} \sum_{p \in P} \frac{1}{c} \mathbb{E}_{t,u}[\mathbf{1}\{f(p) - U_t(p) \geq 0\}(f(p) - U_t(p))]
\end{aligned}$$

Now, in the untruncated distribution, $\mathbb{E}_{t,u}[f(p) - U_t(p)] = -\beta_t^{1/2} \sigma_t^2(p)$, which is negative. Thus, using standard tail bounds Russo and Van Roy (2014),

$$\begin{aligned}
\sum_{t=1}^{T} E[p^* f(p^*) - p^* U_t(p^*)] &\leq \frac{1}{c} \sum_{t=1}^{\infty} \frac{\sigma_t(p)}{\sqrt{2\pi}} e^{-\beta/2} \\
&\leq \frac{1}{c} \sum_{t=1}^{\infty} \frac{\sigma_t(p)}{t^2 |P| c^{-1}} \leq 1
\end{aligned}$$

The result follows from combining all the terms.

Figure EC.3 Decay of E_t under Different Distributions



Notes. At each update step of the Gaussian process, E_t is calculated as the expected difference $\mathbb{E}[\mu_t(p^*) - \mathbb{E}_t[f_t(p^*)]]$, where f_t is obtained averaging over 10,000 posterior draws. Results are averaged over 1,000 independent simulations. The solid black lines show the empirical estimates of $\log(E_t)$, while the blue dotted lines display the corresponding linear fits, with equations annotated in each panel.

EC.5 Simulations Using Alternative WTP Distributions

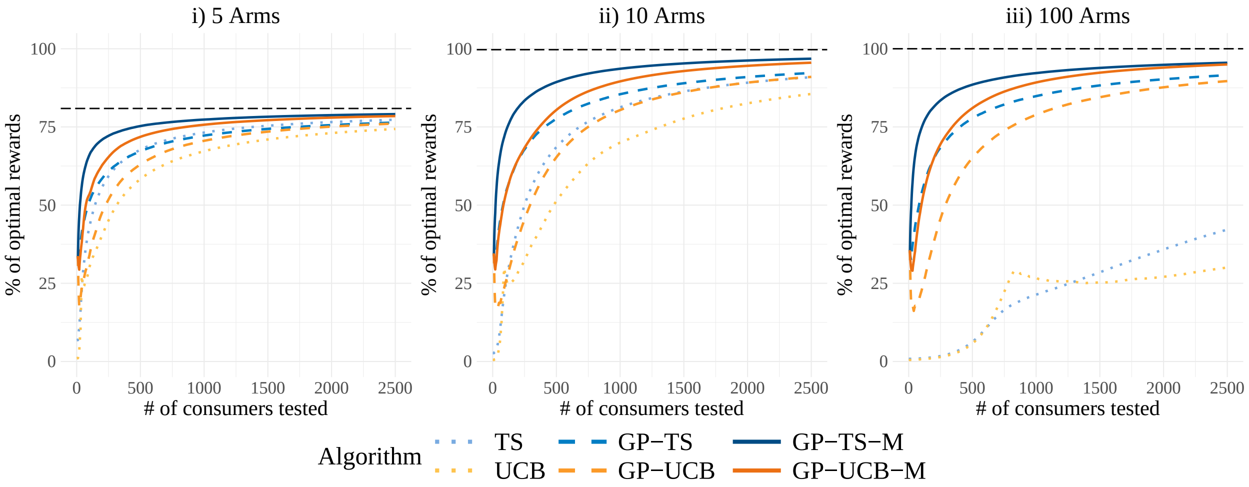
We evaluate the robustness of the proposed method across a variety of challenging scenarios to demonstrate its practical value.

EC.5.1 Field Data

To further demonstrate the applicability of our method, we tested it on real-world data. The data comes from an empirical study of demand for a music streaming subscription service where the distribution of WTP for a monthly plan is estimated (Chou and Kumar 2024; see Figure 2). From this, we then normalize the WTP distribution to lie in the range $[0,1]$; specifically, $p' = \frac{p}{1000}$. Using this WTP distribution, we run the bandit algorithms using the same setup as in the simulations.

Since the optimal price is relatively low (0.21) within the price set, we expect similar uplifts from the informational externalities as observed in the Beta(2,9) case from the main analysis. For the first informational externality, the uplift increases with the number of arms, being slightly negative for 5 arms, slightly positive for 10 arms, and dramatically positive for 100 arms. For the second informational externality, across all price sets the uplifts are positive and similar in size. Additionally, GP-TS-M is the best performing algorithm. These findings are consistent with the Beta(2,9) simulations and highlight the validity and practical value of our method.

Figure EC.4 Field Data: Cumulative Percent of Optimal Rewards (Profits)



Notes. The lines represent the means of the cumulative expected percentage of optimal rewards across 1000 simulations. The black horizontal line represents the maximum obtainable reward given the price set, while 100% represents the true optimal reward given the underlying distribution.

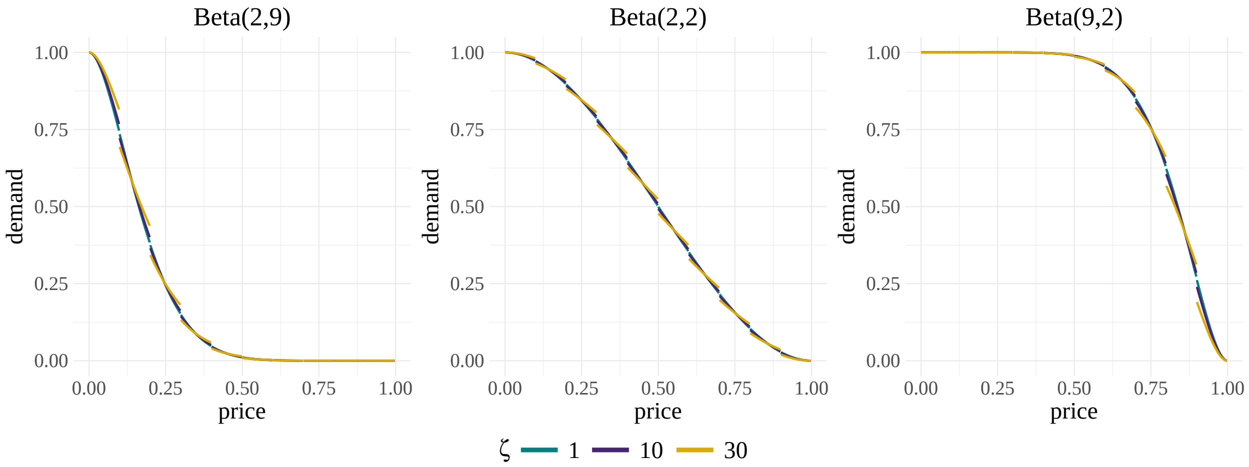
EC.5.2 Discontinuous Demand: Left-Digit Bias

We next discuss the case where consumers may be affected by left-digit bias. This phenomenon occurs when the demand function is discontinuous at a price where the left digit changes (e.g., from \$1.99 to \$2.00), as consumers perceive the price increase to be larger than one cent (Thomas and Morwitz 2005). For instance, Strulov-Shlain (2023) empirically found that consumers reacted to a one-cent increase from a ninety-nine-ending price (resulting in a left-digit change) as if it were a twenty-cent increase. This is an important case to test, as GP-based models rely on continuity and may struggle with discontinuities.

To be consistent with left-digit literature, we discretize the continuum of prices $[0,1]$ into 1000 points and introduce discontinuities at changes in the left-most significant digit (e.g., 0.099 to 0.100, 0.199 to 0.200, etc.). This ensures the left-digit effect occurs between prices ending in ninety-nine and zero. To specify the size of the discontinuities, we calculate the difference in demand between consecutive prices where the left-digit changes and multiply it by a scale factor ζ. For instance, if $\zeta = 20$, the gap is 20 times larger than usual – consistent with Strulov-Shlain (2023). We then rescale the continuous portion of the demand curve to accommodate these gaps.

For our simulations, we used demand curves (Figure EC.5) derived from three WTP distributions – Beta(2,9), Beta(2,2), and Beta(9,2) – adjusted with $\zeta = 10$ (low left-digit bias) and $\zeta = 30$ (high left-digit bias). These cases provide generous bounds for estimates of left-digit bias based on Strulov-Shlain (2023).

Figure EC.5 Left-Digit Demand Curves



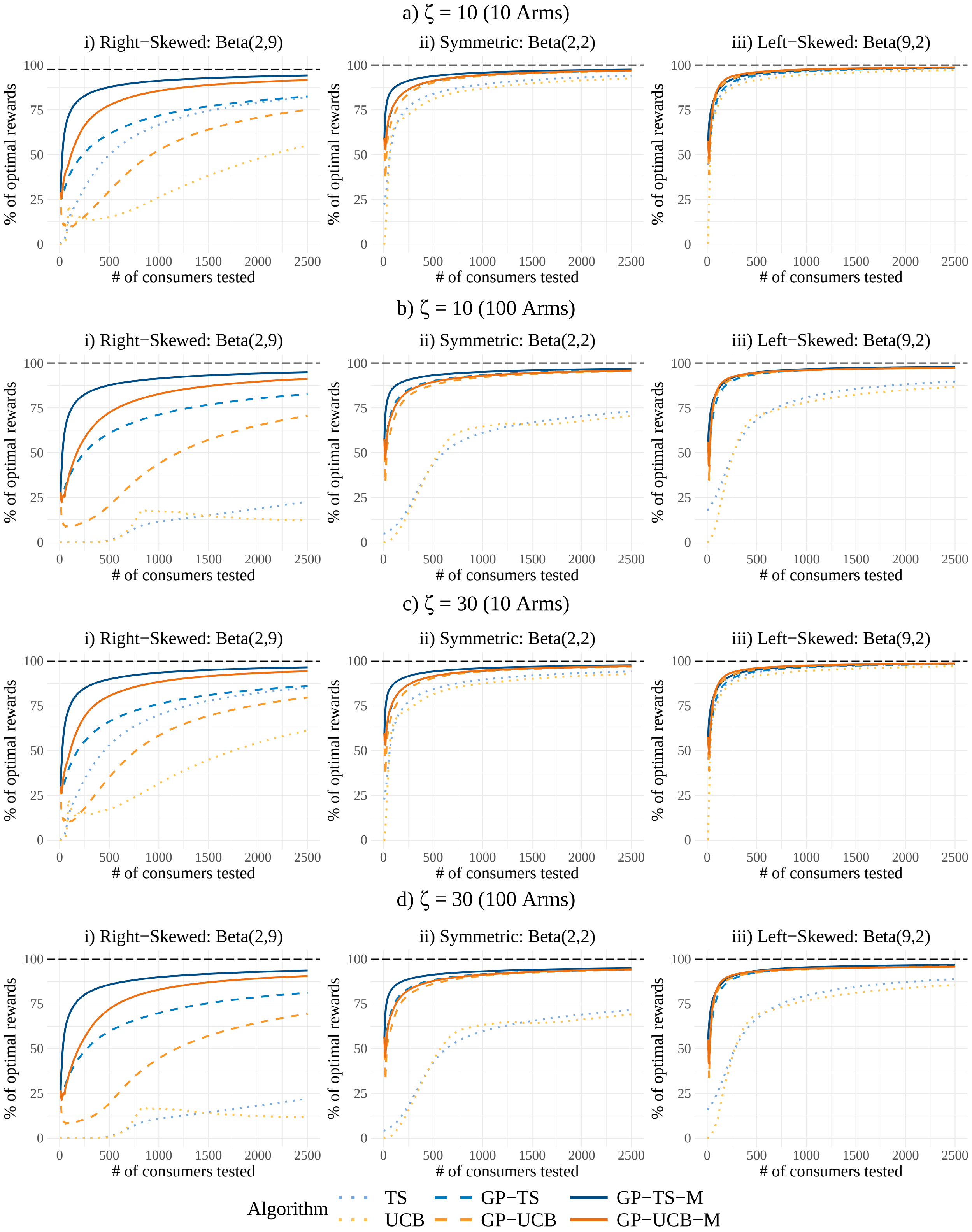
Notes. Price is discretized at intervals of 0.001 from 0 to 1, with left-digit discontinuities occurring every 0.01. $\zeta = 1$ represents the base case with no left-digit bias (discontinuities arise solely from price discretization). $\zeta = 10$ corresponds to low left-digit bias, with a discontinuity gap 10 times larger than the base case. $\zeta = 30$ corresponds to high left-digit bias, with a discontinuity gap 30 times larger than the base case.

Our simulation results are shown in Figure EC.6. With 10 arms, we observe that in both cases (low ζ and high ζ), GP-based algorithms perform as well as in the main analysis. That is with just 10 arms, left-digit discontinuities do not affect GP-based algorithms. Perhaps surprisingly, this outcome is expected as when only 10 prices are tested, each price falls within a distinct interval of the piecewise function, leaving enough distance between prices for the discontinuities to have no impact.

The issue arises when multiple prices are tested within each interval, requiring the GP to learn both the continuous segments and the discontinuity gaps. Since the GP assumes continuity, it tends to smooth over these gaps, resulting in a slight misestimation of the demand curve. For example, in the Beta(2,2) case, when there are low left-digit discontinuity gaps ($\zeta = 10$), GP-TS-M achieves 97.4% (after 2500 consumers) of the true optimal for 10 arms and 96.9% for 100 arms – a slight decrease. However, when there are high left-digit discontinuity gaps ($\zeta = 30$), this decline is more pronounced (97.7% to 95.0%). This is because as the GP tries to smooth over larger discontinuity gaps, it leads to greater misestimation and the selection of slightly suboptimal prices. This can be seen in subfigure d)ii) of Figure EC.6, where the cumulative optimal rewards curve for GP-TS-M flattens before reaching the optimal, as the algorithm prefers to select 0.43, which provides 3% less reward than the true optimal price of 0.39.

Despite the GP's difficulty in handling discontinuities with 100 arms, the monotonic versions of the algorithm remain the best performers after 2500 consumers. While TS and

Figure EC.6 Left Digit: Cumulative Percent of Optimal Rewards (Profits)



a) ζ = 10 (10 Arms)

b) ζ = 10 (100 Arms)

c) ζ = 30 (10 Arms)

d) ζ = 30 (100 Arms)

Notes. The lines represent the means of the cumulative expected percentage of optimal rewards across 1000 simulations. The black horizontal line represents the maximum obtainable reward given the price set, while 100% represents the true optimal reward given the underlying distribution. ζ is a measure of the size of the discontinuity gap that occurs at locations where the left-digit changes.

UCB (which can handle discontinuities as they model each arm independently) will eventually learn the optimal price and surpass the performance of GP-TS-M and GP-UCB-M, this will only happen for extremely large customer counts. For any reasonable number of tested consumers, the additional exploration cost from forgoing the informational externalities far outweighs the small performance loss from slight misestimation.

To summarize, even with left-digit bias, our algorithms continue to perform best empirically. From a managerial perspective, if a firm suspects left-digit bias, a practical approach is to test prices only at the discontinuities (e.g., 1.99, 2.99, etc.) to avoid misspecification issues with the GP. However, testing fewer prices may lower the maximum reward obtainable from the price set. Thus, the trade-off in determining the number of prices to test involves balancing the potential for a higher possible maximum reward against the risk of misestimation. This trade-off is evident in our experiments: in the Beta(2,9) case, when 100 prices were tested instead of 10, GP-TS-M performed better when left-digit bias was low [subfigure a)i) vs. b)i)] but worse when left-digit bias was high [subfigure c)i) vs. d)i)].

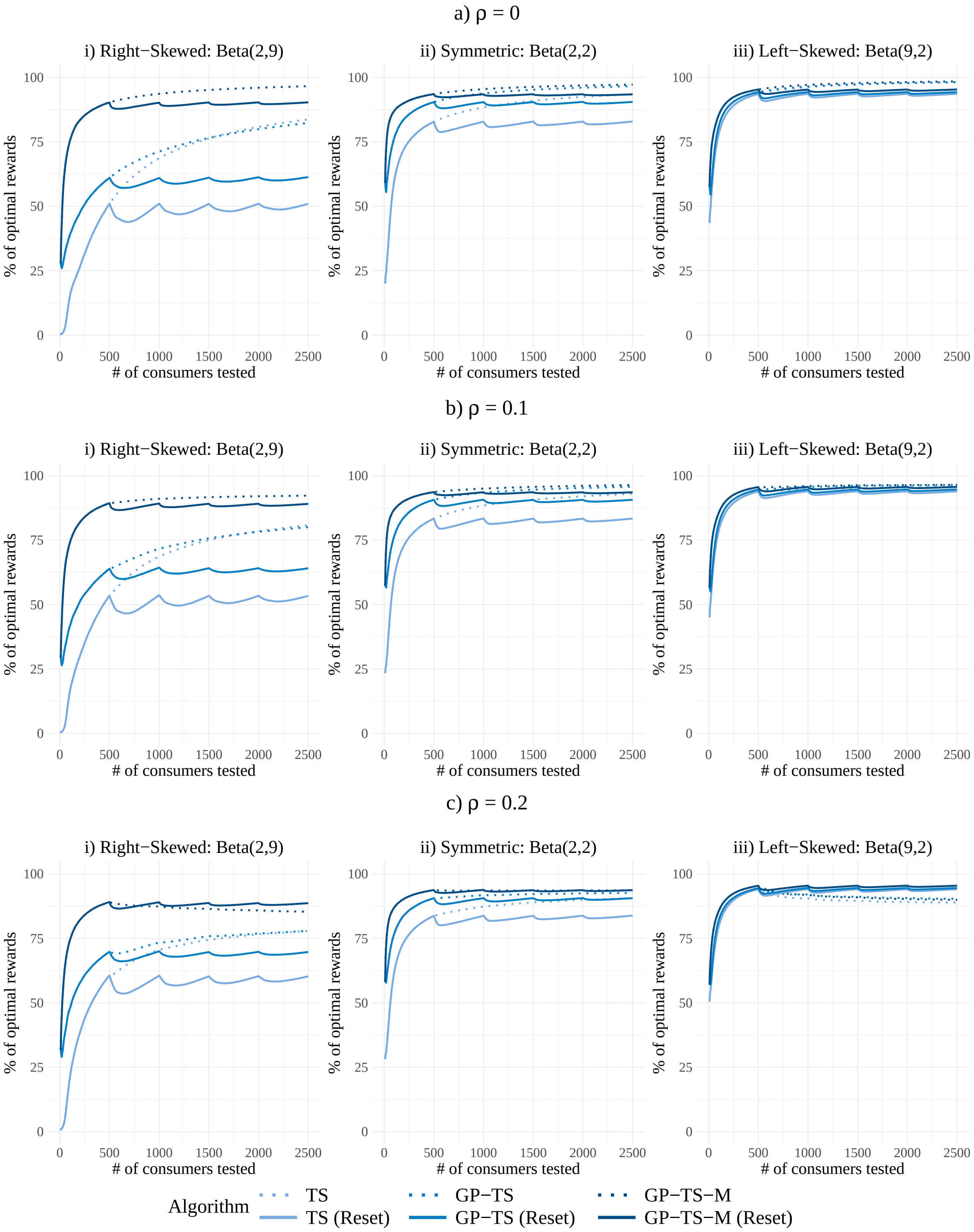
EC.5.3 Time Varying Demand

We now consider the case where demand changes depending on the season (Soysal and Krishnamurthi 2012). We model this by introducing a shock (drawn from the uniform distribution $[-\rho, \rho]$) to the underlying WTP distribution every Q consumers. Specifically, the WTP distribution is shifted horizontally by the value of the shock, increasing or decreasing every consumer's valuation by this amount and resulting in a horizontal shift in the demand curve.

We evaluate the performance (results are shown in Figure EC.7) of the proposed algorithms under these time-varying distributions in two ways. First, we consider the usual baselines (represented by dotted lines), which retain all data from the experiment. Alternatively, we analyze reset variants (represented by solid lines), where the learning process is reset every 500 consumers by discarding all previous experiment data.

First, we analyze how performance changes as ρ varies. When there are no seasonal shocks ($\rho = 0$), resetting results in worse performance for all algorithms. However, the performance gap between the baseline and its corresponding reset variant decreases as informational externalities are incorporated. Furthermore, as ρ increases, the performance decrease from using the reset variants diminish, with GP-TS-M reset variants actually outperforming the baselines when ρ is sufficiently large ($\rho = 0.2$).

Figure EC.7 Time Varying: Cumulative Percent of Optimal Rewards (10 arms)



Notes. The lines represent the means of the cumulative expected percentage of optimal rewards across 1000 simulations. A demand shock is drawn from $[-\rho, \rho]$ every 500 consumers.

Next, we analyze how performance differs across distributions. The performance decrease from using the reset variants instead of the baselines are most pronounced for Beta(2,9) and decrease as the distribution shifts to Beta(2,2) and then to Beta(9,2). In the Beta(9,2) case, when demand shocks are sufficiently large ($\rho = 0.2$), the reset variants outperform the baselines for all three algorithms. However, for the same ρ under Beta(2,9), the TS and GP-TS reset variants perform significantly worse than the baselines, with only GP-TS-M performing better with a reset.

These patterns can be understood by considering the trade-off of resetting. Resetting incurs a cost in learning, as the algorithm must re-learn the demand from scratch. Conversely, resetting enables more accurate learning when the underlying demand has shifted. Thus, resetting performs better only when the losses from learning a new demand curve are less than the losses from using pre-shock data. This suggests that resetting is most applicable when demand shocks are large (i.e., ρ is large) and when learning is relatively easy (i.e., when the optimal price is high within the price set, as in Beta(9,2)).

Finally, regardless of the size of the shocks or the underlying demand curve, resetting is consistently more effective for algorithms with higher learning efficiency. Since incorporating informational externalities enhances learning efficiency, these externalities provide a persistent advantage to the algorithms that leverage them.

EC.6 Heteroscedastic Noise

As discussed in the main results, when there are very few arms, TS can outperform GP-TS. This is intuitive, as the value of the first informational externality (learning across arms) decreases when there are fewer arms that are further apart. Another key consideration is that TS learns the noise separately for each arm, whereas standard GPs assume homoscedastic noise across all arms.

This assumption is a limitation of standard GPs, as the noise around purchase rates is inherently heteroscedastic. Intuitively, the sample mean at a price where nearly every consumer either purchases or does not purchase is much less noisy than at a price where consumers are equally likely to purchase and not purchase. Specifically, since purchase is a binary decision, the variance at purchase rate y_{at} is given by $y_{at}(1 - y_{at})$. This appendix addresses this limitation by introducing an alternative method for tuning the noise hyperparameter to account for heteroscedasticity.

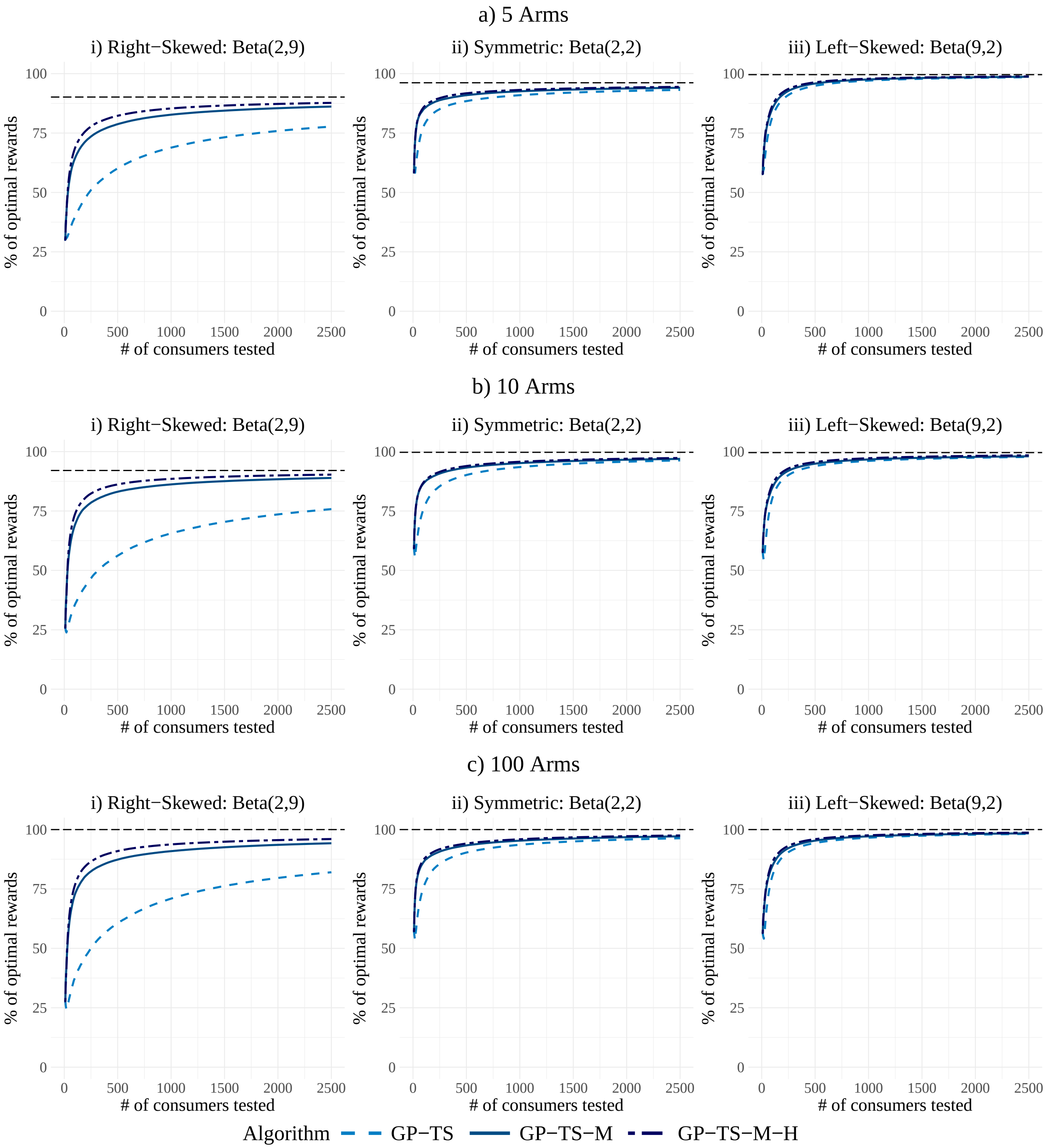
Implementation: While there are methods to estimate a GP with heteroscedastic noise (Goldberg et al. 1997, Kersting et al. 2007), they face the same challenges as estimating noise using MLE under the homoscedastic specification. Specifically, it can be difficult to accurately identify the shape and noise hyperparameters (Murray 2008), which poses significant problems for bandits, as an insufficient noise estimate can cause the algorithm to get stuck on suboptimal arms.

An alternative approach is to model the error by specifying an underlying structural process. In our case, the noise can be modeled based on how likely a consumer is to purchase. Generally, this can be written as $\sigma_y^2 = g(D(p))$ for some unknown function g. However, because the noise depends on the underlying distribution, which is completely unknown, it is unlikely that any suitable candidates for $g(\cdot)$ exist.

Our estimation process for heteroscedastic noise operates as follows. First, we estimate the GP using homoscedastic noise, producing the standard results. Next, we take a demand draw from this GP, which (due to the binary nature of purchase decisions) allows the noise estimate at each price to be calculated as $\tilde{D}(p)(1-\tilde{D}(p))$. Using this noise input, we estimate another GP, after which the bandit process continues as usual. Importantly, as this method updates with a new noise draw at each update iteration, it provides more accurate noise estimates while still preventing the algorithm from getting stuck from consistently underestimating noise. We refer to this implementation as *GP-TS-H*, with its monotonic version called *GP-TS-M-H*.

Results: The results are presented in Figure EC.8. Including heteroscedasticity in addition to monotonicity results in a small performance increase across all simulations. As with other informational externalities, the effects are most pronounced in the Beta(2,9) case. This is because smaller noise hyperparameters synergize with monotonicity to reduce the space of potential demand curves in the low-reward, high-price region.

Figure EC.8 Heteroscedasticity: Cumulative Percent of Optimal Rewards (Profits)



Notes. The lines represent the means of the cumulative expected percentage of optimal rewards across 1000 simulations. The black horizontal line represents the maximum obtainable reward given the price set, while 100% represents the true optimal reward given the underlying distribution.