Journal of Machine Learning Research 27 (2026) 1-58 Submitted 1/25; Published 3/26

Impatient Bandits: Optimizing for the Long-Term Without Delay

Kelly W. Zhang KELLY.ZHANG@IMPERIAL.AC.UK
Imperial College London

Thomas Baldwin-McDonald* TOMMCDONALD955@GMAIL.COM
University of Manchester

Kamil Ciosek KAMILC@SPOTIFY.COM
Spotify

Lucas Maystre† LUCAS@REFLECTION.AI
Reflection AI

Daniel Russo DJR2174@GSB.COLUMBIA.EDU
Columbia University

Editor: Tor Lattimore

Abstract

Increasingly, recommender systems are tasked with improving users' long-term satisfaction. In this context, we study a content exploration task, which we formalize as a bandit problem with delayed rewards. There is an apparent trade-off in choosing the learning signal: waiting for the full reward to become available might take several weeks, slowing the rate of learning, whereas using short-term proxy rewards reflects the actual long-term goal only imperfectly. First, we develop a predictive model of delayed rewards that incorporates all information obtained to date. Rewards as well as shorter-term surrogate outcomes are combined through a Bayesian filter to obtain a probabilistic belief. Second, we devise a bandit algorithm that quickly learns to identify content aligned with long-term success using this new predictive model. We prove a regret bound for our algorithm that depends on the *Value of Progressive Feedback*, an information-theoretic metric that captures the quality of short-term leading indicators that are observed prior to the long-term reward. We apply our approach to a podcast recommendation problem, where we seek to recommend shows that users engage with repeatedly over two months. We empirically validate that our approach significantly outperforms methods that optimize for short-term proxies or rely solely on delayed rewards, as demonstrated by an A/B test in a recommendation system that serves hundreds of millions of users.

Keywords: bandit algorithms, delayed rewards, progressive feedback, recommender systems

1. Introduction

The multi-armed bandit problem stands as a cornerstone in machine learning, statistics, and operations research, crystallizing the challenge of learning to make effective decisions through

∗. Research done as part of an internship at Spotify.
†. Research done while at Spotify.

intelligent trial and error. In its classical formulation, the model assumes immediate reward observation following an action, facilitating rapid adaptation. However, this instantaneous feedback structure often fails to capture the complexity of real-world applications, particularly in the realm of digital platforms serving millions or billions of users.

On these large-scale platforms, algorithms need to make decisions at a rapid pace, since the time between the start of one user's interaction and the start of the next user's interaction is extremely brief. Consequently, decision-making algorithms that optimize for rewards that are observed before the start of the subsequent user interaction (or even before a few interactions) inevitably skew towards extremely short-term goals, such as whether a user clicked on an advertisement. Selecting longer-term outcomes for the reward (like long-term engagement with a recommendation) better aligns with genuine long-term goals. However, waiting even a brief amount of time to observe the reward following each decision can greatly hinder the efficacy of bandit algorithms, since these brief delays can correspond to waiting an enormous number of interactions or "rounds" in classical bandit formulations. In Figure 1, we depict this apparent trade-off between speed of learning versus the alignment of the reward with the long-term outcome of interest.

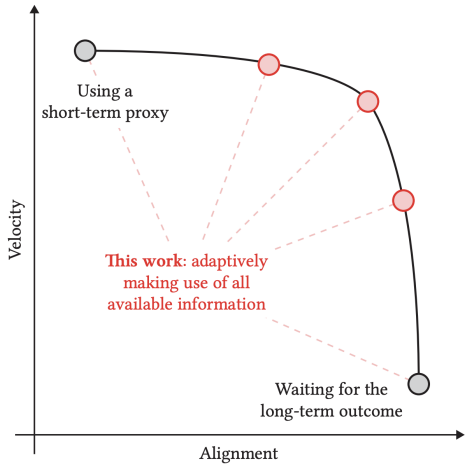

Figure 1: Short-term outcomes enable a rapid feedback loop, but might be poorly aligned with long-term success metrics, which take longer to realize. Our method bypasses this apparent trade-off by adaptively making use of all available information at a given time.

In this work, we investigate the problem of optimizing for true long-term goals while mitigating the impact of delay. Our key insight is that most long-term outcomes become increasingly predictable over time. In the context of digital platforms, a user's long-term engagement is revealed incrementally. For instance, total spending is signaled by initial purchases, and the likelihood of conversion following a marketing message diminishes as time passes without a response. We term this phenomenon "progressive feedback," distinguishing it from the widely studied delayed feedback, where no information is gleaned until long after an action is taken.

Our interest in this problem stems from a real challenge encountered at Spotify, a leading audio streaming service. Following a content recommendation decision, engagement feedback from each user is progressively revealed; we illustrate examples of the progressive engagement feedback in Figure 2. Recent efforts to optimize for longer-term metrics, such as user engagement over a 60-day period, brought substantial benefits to the recommendation system, as compared to optimizing for shorter-term outcomes (Maystre et al., 2023; Wang et al., 2022; Zou et al., 2019; Yang et al., 2024). However, these long-term metrics are only realized after a significant delay, meaning such data is never available for recently released content.

In this context, we focus on the cold-start problem—the challenge of rapidly learning to make effective recommendations for newly released content with little to no historical

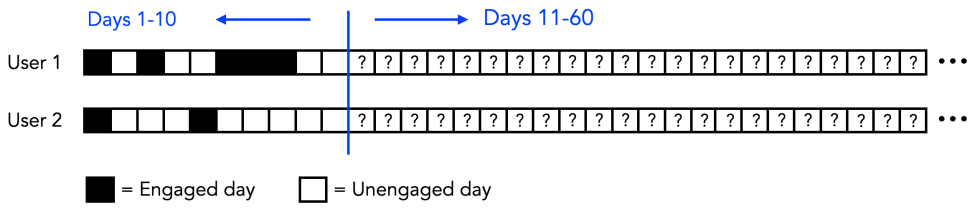

Figure 2: Progressive engagement feedback from two hypothetical users. The long-term outcome of interest is users' average engagement over the 60 days following a recommendation. Each user's long-term outcome is increasingly predictable as more engagement information is revealed.

data. The core challenge lies in optimizing for the long-term outcome and identifying content with superior long-term performance, but doing so without waiting for these long-term metrics to become fully revealed. We address this problem in a novel way, by training a (empirical) Bayesian filtering model that draws inferences about content quality and appeal from progressively revealed user engagement. A/B tests have shown that incorporating progressive feedback has large benefits in practice (see Section 8), and this idea is now used in production as a core part of the recommender system at Spotify, which powers personalized audio recommendations for hundreds of millions of users.

This paper develops a general model of multi-armed bandit problem with progressive feedback, which distills the core challenge of the problem encountered at Spotify. We discuss the contributions of the paper in detail in the next subsection.

1.1 Contributions

This work expands upon the algorithm originally presented in McDonald et al. (2023). Key additions include *a*) a formalization of the model, *b*) a novel regret analysis, *c*) A/B test results from an implementation of the algorithm at Spotify, and *d*) several additional synthetic experiments.

Model of bandits with progressive feedback. Motivated by this real-world problem faced at Spotify and recognizing its broader relevance, we formulate a model termed *bandits with progressive feedback* (Section 3). In this setting, the reward from selecting an "arm" is fully observed only after a significant delay but becomes progressively more predictable over time. We primarily focus on a key special case where rewards are a linear function of a set of correlated Gaussian engagement outcomes. The crux of the problem lies in the interplay between the delay structure, where some outcomes are revealed much earlier than others, and the correlation structure, which determines the extent to which initial observations are predictive of later ones. In our problem setting, the algorithm confronts two sources of limited feedback. The first is *bandit feedback* (in contrast to *full feedback*), where the algorithm only has data on items it has recommended in the past, necessitating active exploration for learning. The second is *progressive feedback*, where engagement outcomes are revealed gradually over time, introducing delay in the resolution of uncertainty regardless of the number of users receiving recommendations.

Impatient bandit algorithm. To address the above challenges within our model, we propose an algorithm that integrates two key components: Thompson sampling and a (empirical) Bayesian filter (Section 4). Thompson sampling (Thompson, 1933) is employed to tackle the bandit feedback challenge. The Bayesian filter is designed to mitigate the issues arising from progressively revealed feedback. The Bayesian filtering model projects the true average reward of each based on partially observed engagement trajectories, maintaining appropriate residual uncertainty. Both mean and uncertainty estimates are incrementally updated as further feedback is observed. Our model assumes this filter is computed based on highly informed prior beliefs. In practice, this rich prior comes from historical data, by training the hyperparameters of the Bayesian filter on complete trajectories of user interactions with historical content (Section 6).

Regret analysis and the value of progressive feedback. We derive a novel regret bound for this algorithm that provides rigorous insight into the value of progressive feedback (Section 5). When there is no useful progressive feedback, our regret bound reduces to a standard bound for Thompson sampling with delayed feedback. With progressive feedback, our regret bound improves based on what we term the *value of progressive feedback*. This term equals the mutual information between censored (i.e., delayed) user-engagement trajectories and the true reward observations that will eventually materialize. Note that this mutual information term is entirely a property of the underlying data generating environment itself, and is not impacted by the decision-making algorithm. High mutual information indicates that early feedback substantially resolves uncertainty in long-term rewards.

Importantly, our focus differs from traditional bandit analyses that emphasize how quickly regret vanishes as the number of arm selections grows. In settings with severely delayed rewards, such as our motivating application where the primary reward metric is computed after 60 days, it is critical to learn something meaningful before *any* rewards are fully observed. This is challenging regardless of the number of arm selections made during that time frame. Our theoretical and practical objective, therefore, is to take advantage of progressive feedback to significantly improve decision quality compared to scenarios without such feedback. Our algorithm achieves this: We prove that it incurs low expected regret in any batch when the value of progressive feedback is high, even before long-term rewards are observed.

Semi-Synthetic Experiments and A/B Test Results. Experiments with a synthetic data generating process show that our theory aligns qualitatively with performance across a range of problems with varying true "values of progressive feedback" (Section 7). We also include experiments that utilize podcast recommendation data from Spotify; these simulation results suggest that there can be substantial value in incorporating progressive feedback in real life applications. Additionally, we present results from a large-scale A/B test that was run at Spotify (Section 8). The A/B test compares a recommendation algorithm that incorporated progressive feedback to one that did not. The results illustrate the significant benefit that leveraging progressive feedback can have in complex, industrial-scale recommender systems.

2. Related Work

Surrogate Outcomes. The *surrogate outcomes* literature uses short-term outcomes as a proxy for long-term outcomes. For example, for a treatment aimed at preventing heart

disease, changes in cholesterol level may be used as a surrogate outcome for the distal outcome of developing heart disease in the next year. Most of this literature focuses on estimating causal effects, rather than decision-making. There are three general approaches to using surrogate outcomes: *a*) assume a causal relationship between the surrogate and long-term outcome, and effectively replace the distal outcome with surrogates (Athey et al., 2016; Duan et al., 2021; Prentice, 1989; Fleming and Powers, 2012; Weintraub et al., 2015; VanderWeele, 2013), *b*) use surrogates that are predictive of the long-term outcome to improve imputation of missing distal outcomes (Kallus and Mao, 2020; Cheng et al., 2021), and *c*) use surrogates that are predictive of the long-term outcome to fit a joint Bayesian model of these outcomes (Tripuraneni et al., 2024; Richardson et al., 2023; Anderer et al., 2022). Our work is most similar in spirit to that last approach, with a focus on decision-making.

Optimizing for Long-Term Outcomes in Recommendation Systems. In the recommendation systems literature, there are two foremost approaches to designing algorithms that optimize for long-term outcomes. The first focuses on designing algorithms that account for the delayed effects of actions on the same users over time, e.g., by moving beyond bandit algorithms and designing algorithms that account for how a recommendation decision today may affect that same user's future engagement (Xu et al., 2023; Wu et al., 2017; Zheng et al., 2018; Ji et al., 2021). The second approach involves designing algorithms that optimize for long-term, delayed rewards and incorporate short-term outcomes in some way to speed up learning, e.g., by using the short-term outcome as a surrogate or combining short-term and long-term outcomes in the definition of the reward (Wang et al., 2022; Zou et al., 2019; Liu et al., 2024; Zhang et al., 2021); None of these existing works focus on the cold-start recommendation setting in which there is little to no historical data for certain actions (the focus of this work). With the exception of Zhang et al. (2021), which we discuss below, these existing works develop offline RL methods, which use large datasets in which each action has been played many times already.

Online Decision-Making Algorithms that use Intermediate Outcomes. Delayed rewards are a significant challenge in online decision-making settings, and many bandit algorithms have been developed specifically to accommodate such delays (Howson et al., 2023; Mandel et al., 2015; Desautels et al., 2014; Bistritz et al., 2019; Thune et al., 2019; Vernade et al., 2020; Pike-Burke et al., 2018; Joulani et al., 2013; Zhou et al., 2019). We approach the delayed rewards problem by incorporating intermediate, surrogate outcomes that may be predictive of the reward. Several previous works also consider algorithms with access to intermediate outcomes, but consider different problem settings and/or make different assumptions on the type of intermediate feedback.

Anderer et al. (2022) use surrogate outcomes to design a Bayesian adaptive clinical trial which adaptively decides whether to stop the trial at a pre-specified decision point, subject to a power constraint. Wu and Wager (2022) develop a Thompson Sampling algorithm for survival outcomes based on the proportional hazards model, which leverages incrementally revealed information on how long individuals have survived so far. Grover et al. (2018) consider a UCB algorithm for best arm identification in a setting with stochastically delayed rewards in which the next decision cannot be made until the delayed reward is observed; this delay structure differs fundamentally from ours. Moreover, they assume that their algorithm sees i.i.d. intermediate outcomes (which may be biased or unbiased estimates of the mean

reward) while it waits, which does not accommodate progressively revealed user engagement feedback (see Figure 2), which critically are dependent over time.

Yang et al. (2024) apply a surrogate index approach (Athey et al., 2016) to develop a Thompson sampling algorithm that optimizes a fitted function of short-term outcomes instead of the true long-term reward (customer retention). While they assume short-term outcomes act as perfect surrogates for the long-term reward, our approach incorporates *imperfect* leading indicators. Finally, Zhang et al. (2021) use importance weighting to ensure the short-term surrogate is an unbiased estimate of the mean, long-term reward. Their algorithm (developed for a setting with binary outcomes and a single intermediate outcome) does not easily extend to settings with many intermediate outcomes (> 50 in our applications) that are continuous, i.e., where estimating propensities to form importance weights is more difficult.

Distinguishing features of our work. Our work advances the literature in several key ways. First, unlike most existing models which treat feedback as either delayed or immediate, or assume perfect surrogate outcomes, we formalize a model of *progressive feedback* where information becomes incrementally more predictive over time. Our model addresses the practical challenge of synthesizing information from many (> 50) sequentially revealed intermediate signals that serve as imperfect leading indicators rather than perfect surrogates of true rewards. Second, we introduce a novel Gaussian filtering algorithm specifically designed to synthesize information from numerous progressively revealed outcomes. This approach not only predicts rewards but crucially maintains appropriate uncertainty estimates throughout the learning process, preventing premature convergence to suboptimal decisions. While this uncertainty-aware approach shares some features with survival modeling work (Wu and Wager, 2022), our method uniquely handles multiple continuous intermediate signals. Third, we introduce a novel information-theoretic approach to quantifying progressive feedback value and prove novel regret bounds based on this measure. Finally, our approach has demonstrated significant impact within a large-scale industrial recommender system, validating its practical effectiveness in handling the complexities of real-world progressive feedback at scale.

3. Problem Formulation

We now formally describe our problem setup, which is motivated by recommender systems applications. In our problem setup, a large batch of actions (items) are selected at each decision time. Specifically, each decision time $t \in \mathbb{N}$ represents a day during which many users, denoted by the set $\mathcal{U}_t$, are simultaneously presented with item recommendations. Note that the sets $\{\mathcal{U}_t\}_{t\in\mathbb{N}}$ are disjoint and we use $\mathcal{U} \triangleq \cup_{t\in\mathbb{N}} \mathcal{U}_t$ to denote the collection of all users. For simplicity, we assume that the batches $\mathcal{U}_t$ are of the same size each day; We use $m = |\mathcal{U}_1| = |\mathcal{U}_2| = |\mathcal{U}_3| = \ldots$ to denote the batch size. While batching is common in practice for operational reasons, we employ it primarily as a device to clearly distinguish between two crucial aspects of the problem. The first aspect is the passage of time, which is represented by the number of batches. The second is the rate at which exploratory recommendations are made, which is determined by the size of each batch. This model enables us to separately vary the pace of decision-making from the pace at which outcomes are revealed (amount of delay).

Each day $t \in \mathbb{N}$, for each user $u \in \mathcal{U}_t$ in the batch, the system selects an item A_u from a finite collection of items $\mathcal{A}$ to recommend ($\mathcal{A}$ is the action space). Each item $a \in \mathcal{A}$ has an associated item feature vector $Z_a \in \mathcal{Z}$ for $|\mathcal{Z}| < \infty$. For example, in the podcast recommendation setting, Z_a may include the category of the podcast show (comedy, news, lifestyle). The item features are used as they may be predictive of or correlated with the item's performance. Note that this vector is not a typical context or state vector, rather it represents features of the *item* (action). Note that we are able to extend the algorithm we present to settings with user contexts (see Appendix C for details); throughout the main body of this paper we focus on the setting without context for clarity.

Upon recommending an item $A_u \in \mathcal{A}$ for a user $u \in \mathcal{U}_t$, the algorithm subsequently observes a collection of user outcomes $Y_u^{(1)}, Y_u^{(2)}, \ldots, Y_u^{(J)}$. We use $Y_u \triangleq (Y_u^{(1)}, Y_u^{(2)}, \ldots, Y_u^{(J)}) \in \mathbb{R}^J$ to refer to the entire vector of engagement outcomes observed from selecting action A_u. In our setting, these outcomes are progressively revealed over time. Specifically, in our application, $\left(Y_u^{(j)}\right)_{j=1}^J$ refer to measures of engagement with the recommended item that are revealed over J days. An engagement trajectory is associated with a long-term reward $R(Y_u)$ by applying a fixed, known, real-valued, reward function $R(\cdot)$. For example, the reward could represent the total user engagement over time, $R(Y_u) = \sum_{j=1}^J Y_u^{(j)}$, or if the outcome of interest is the final engagement, one could choose $R(Y_u) = Y_u^{(J)}$. In general, the outcomes $\left(Y_u^{(j)}\right)_{j=1}^J$ could be a series of predictions of a long-term user outcome from a previously trained model or simply leading indicators that are correlated with a long-term metric of interest.

Formally, we use d_j to refer to the deterministic "delay" in seeing the j^{th} outcome $Y_u^{(j)}$. For example, if $d_1 = 0, d_2 = 1, \ldots, d_J = J - 1$ and user u is in batch t, then $Y_u^{(1)}$ is revealed immediately after selecting the action A_u at time t. Following decision time $t + 1$, additional engagement information $Y_u^{(2)}$ is observed, and so on until following decision time $t + J - 1$ the final engagement outcome $Y_u^{(J)}$ is revealed.

3.1 Counterfactual Outcomes and Learning Objective

We use the potential outcomes framework (Imbens and Rubin, 2015; Rubin, 1974) to formally represent counterfactual outcomes. We assume there exist latent variables

$$\left\{Y_u(a) = \left(Y_u^{(1)}(a), Y_u^{(2)}(a), \ldots, Y_u^{(J)}(a)\right)\right\}_{a \in \mathcal{A}, u \in \mathcal{U}}$$

called *potential engagement outcomes*, such that the engagement outcome of selecting action $A_u = a$ for some $a \in \mathcal{A}$ is $Y_u(a) = \left(Y_u^{(1)}(a), Y_u^{(2)}(a), \ldots, Y_u^{(J)}(a)\right)$; the observed outcome is $Y_u \triangleq Y_u(A_u)$. This means that the potential engagement outcome is a function of only the recommendation that user receives and not the recommendations received by other users, ruling out, for instance, complicated social network effects.[1]

The next assumption formalizes a symmetry of the distribution of outcomes among users in the cohort $\mathcal{U}$.

1. This assumption is called the Stable Unit Treatment Value Assumption (SUTVA) in the causal inference literature. A growing literature on experimentation in the presence of interference tries to make sensible inferences when this assumption is relaxed (Li and Wager, 2022; Han et al., 2023).

Assumption 1 (Exchangeability). *For any fixed item $a \in \mathcal{A}$, the potential outcome vectors $\big(Y_u(a)\big)_{u\in\mathcal{U}}$ are exchangeable over users $u \in \mathcal{U}$ conditional on $Z_a = z$ for any $z \in \mathcal{Z}$. That is, the joint distribution satisfies*

$$\big(Y_u(a)\big)_{u\in\mathcal{U}} \mid (Z_a = z) \quad \stackrel{D}{=} \quad \big(Y_{\sigma(u)}(a)\big)_{u\in\mathcal{U}} \mid (Z_a = z)$$

for any permutation σ over $\mathcal{U}$.

Remark 1 (Interpreting Exchangeability). *To interpret this assumption, it is helpful to think of $\mathcal{U} = \{1, 2, 3 \ldots\}$ as a list of user IDs that are randomly assigned to users in the cohort. Then, because of the random assignment, there is nothing a priori to differentiate the distribution of outcomes for two users who receive the same recommendation at the same time (i.e., $u, u' \in \mathcal{U}_t$ with $A_u = A_{u'}$). This part of the assumption is not a major restriction. The meaningful assumed structure is a symmetry in outcomes among users who receive the same recommendation at different points in time. This is critical to allowing us to learn from initial recommendations how to make recommendations to future batches of users.*

By De Finetti's theorem (Heath and Sudderth, 1976; De Finetti, 1937), the exchangeability assumption is equivalent to assuming that there exists a random, latent (i.e., unobserved) variable θ_a such that the user's potential engagements $\big\{Y_u(a)\big\}_{u\in\mathcal{U}}$ are i.i.d. given θ_a, Z_a. This means there exists a distribution P_{θ_a, Z_a} such that

$$Y_1(a), Y_2(a), Y_3(a), \cdots \mid \theta_a, Z_a \stackrel{\text{i.i.d.}}{\sim} P_{\theta_a, Z_a}.$$

While Z_a captures an item's observed features, we can think of θ_a as reflecting latent features of the item—like its quality and style—which cannot be inferred prior to recommending it. Intuitively, user responses $Y_u(a)$ are not i.i.d. (only *conditionally* i.i.d.) because observing $Y_u(a)$ gives one more information about the latent θ_a, which reduces uncertainty in $Y_{u'}(a)$ for some user $u' \neq u$.

Recall that the reward $R(Y_u)$ is a fixed, known function of the engagement trajectory Y_u. Define the expected reward from deploying an item a in the population by

$$\bar{R}_a \triangleq \mathbf{E}\left[R\big(Y_u(a)\big) \mid Z_a, \theta_a\right]. \tag{1}$$

The above expectation averages over the randomness in $Y_u(a)$ given Z_a and θ_a, which determine the outcome distribution for that item, P_{θ_a, Z_a}. We define the reward-maximizing item by

$$A^* \triangleq \arg\max_{a\in\mathcal{A}} \{\bar{R}_a\}.$$

If the best item is not unique, we simply set A^* to be one of the optimal items in $\mathcal{A}$.

A *policy* π is a (possibly randomized) rule that at each decision time t makes recommendation decisions A_u to the users in batch t as a function of the history of observations available to it, $\mathcal{H}_t$. Due to delayed feedback, we need to develop some extra notation to define $\mathcal{H}_t$ precisely. We use $\tau_u \in \mathbb{N}$ to refer to the decision time at which the user $u \in \mathcal{U}$

receives their recommendation. That is, $u \in \mathcal{U}_{\tau_u}$. Using this notation, we define a *censored* version of the potential engagement outcome $Y_u(a)$ as follows:

$$\tilde{Y}_{u,t}^{(j)} \triangleq \begin{cases} Y_u^{(j)} & \text{if } t > \tau_u + d_j, \\ \text{Null} & \text{otherwise.} \end{cases} \tag{2}$$

Define $\tilde{Y}_{u,t} \triangleq (\tilde{Y}_{u,t}^{(1)}, \tilde{Y}_{u,t}^{(2)}, \ldots, \tilde{Y}_{u,t}^{(J)})$. At decision time t, the system must base its decisions on $\{Z_a\}_{a\in\mathcal{A}}$ and on the censored history

$$\mathcal{H}_t \triangleq \bigcup_{u\in\mathcal{U}_{<t}} \left\{ A_u, \tilde{Y}_{u,t}^{(1)}(A_u), \tilde{Y}_{u,t}^{(2)}(A_u), \ldots, \tilde{Y}_{u,t}^{(J)}(A_u) \right\} \cup \{Z_a : a \in \mathcal{A}\}.$$

Above, $\mathcal{U}_{<t} \triangleq \bigcup_{t'=1}^{t-1} \mathcal{U}_{t'}$. That is, the censored history at decision time t contains the activity measurement $Y_u^{(j)}(a)$ if and only if user u was recommended item a more than d_j periods ago. The above implies that $Y_u^{(j)}$ is observed when $t > \tau_u + d_j$. Define the instantaneous regret for a policy π at batch t,

$$\Delta_t(\pi) \triangleq \frac{1}{|\mathcal{U}_t|} \sum_{u\in\mathcal{U}_t} \left\{ R\big(Y_u(A^*)\big) - R\big(Y_u(A_u)\big) \right\}. \tag{3}$$

Our regret bounds will control the "average" regret. Next we discuss exactly what we mean by average regret and how it relates to the cold start problem.

Cold Start and Bayesian Regret. The cold-start problem is faced on a recurring basis by the company (e.g., new content is released each day). When new content is released, we view it as a random draw from some distribution over possible items. This means that when a new item $a \in \mathcal{A}$ is released, we can think of observed and latent item features Z_a and θ_a respectively (which determine P_{θ_a,Z_a}, the response distribution for the item a) as being drawn from some underlying distribution. Assumption 2 below formalizes this.

Assumption 2 (Randomly Drawn Items)**.** *The items in $\mathcal{A}$ (note $|\mathcal{A}| < \infty$) are drawn i.i.d. from a distribution of possible items. Mathematically, this means that the observed and latent item features, $\{Z_a, \theta_a\}_{a\in\mathcal{A}}$, are i.i.d. over $a \in \mathcal{A}$.*

Mathematically, the consequence of Assumption 2 is that the random item response distributions $\left\{P_{\theta_a,Z_a}\right\}_{a\in\mathcal{A}}$ are drawn i.i.d. over $a \in \mathcal{A}$. We call the distribution of latent features θ_a given the observed features Z_a the *prior* distribution for θ_a. In fact the data generating process can be thought of as a hierarchical model wherein for an item $a \in \mathcal{A}$, (i) Z_a is sampled, (ii) $\theta_a \mid Z_a$ is sampled from a prior distribution that depends on Z_a, and (iii) subsequently potential response outcomes are sampled conditionally i.i.d. as $Y_1(a), Y_2(a), \cdots \mid \theta_a, Z_a \overset{\text{i.i.d.}}{\sim} P_{\theta,Z_a}$.

We call the following the expected instantaneous regret (or *Bayesian regret*) for a given policy π in batch t:

$$\mathbf{E}\left[\Delta_t(\pi)\right] = \mathbf{E}_\pi \left[\frac{1}{|\mathcal{U}_t|} \sum_{u\in\mathcal{U}_t} \left\{ R\big(Y_u(A^*)\big) - R\big(Y_u(A_u)\big) \right\} \right] \tag{4}$$

$$= \mathbf{E}\left[\mathbf{E}_\pi \left[\frac{1}{|\mathcal{U}_t|} \sum_{u\in\mathcal{U}_t} \left\{ R\big(Y_u(A^*)\big) - R\big(Y_u(A_u)\big) \right\} \;\middle|\; \{Z_a, \theta_a\}_{a\in\mathcal{A}} \right] \right]. \tag{5}$$

Above, we use $\mathbf{E}_\pi$ to denote the expectation when policy π is used to select actions and the outer expectation in (5) averages over the draw of items $\{Z_a, \theta_a\}_{a\in\mathcal{A}}$. Note that the above expectation averages over the randomness in a) the draw of the observed and latent item features, $\{Z_a, \theta_a\}_{a\in\mathcal{A}}$, which determines the distribution $\{P_{\theta_a,Z_a}\}_{a\in\mathcal{A}}$, b) the draw of potential outcomes $Y_u(a)$ from P_{θ_a,Z_a} given Z_a, θ_a, and c) the recommendation decisions A_u made by the (possibly randomized) policy π.

Our Bayesian approach is not merely a philosophical choice, but a pragmatic one, reflecting the recurring nature of cold-start problems in recommender systems. The expected value in (4) can be interpreted as the long-run regret incurred across many instances of this problem, faced repeatedly as new content is released. This interpretation aligns with the practical reality of recommendation systems, where new items are continually introduced. This recurring structure also motivates our theoretical assumption that the algorithm has access to an informed prior distribution. In practice, we construct this informed prior by leveraging historical item engagement data from previously recommended items not in the current set $\mathcal{A}$; See Section 6. This approach lets us leverage patterns in historical data to inform current recommendations. For instance, we might infer immediately that an item is likely of low quality given its features Z_a, or might infer that an item is likely to offer high long-term reward given the strong recurring engagement patterns users have displayed over a shorter time period.

A common objective, roughly speaking, is to minimize the expected instantaneous regret averaged over T batches:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}[\Delta_t(\pi)]. \tag{6}$$

Our upper bounds are more refined and also bound $\mathbf{E}[\Delta_t(\pi)]$ for *every* batch of users t. Rather than focus on attaining vanishing regret as the number of batches scales, our focus is on achieving fairly low regret after a short timespan (low t or T), well before all rewards are fully observed. In our recommender system application, this translates to an ability to effectively recommend content soon after its release. Our theory confirms that our algorithm can leverage progressive feedback to learn far more rapidly than would otherwise be possible in settings with severely delayed rewards.

3.2 Gaussian distribution

Next, we introduce a Gaussian assumption on the outcomes $Y_u^{(j)}$. This assumption enables computing a posterior distribution in closed form. Additionally, we will prove formal regret bounds under this Gaussian assumption. Note, however, that the algorithm we develop can be generalized beyond Gaussian outcomes.

Assumption 3. *For any action $a \in \mathcal{A}$, the potential outcomes $\big(Y_u(a)\big)_{u\in\mathcal{U}}$ are a Gaussian process. That is, for any finite subset of $\mathcal{U}' \subset \mathcal{U}$, the distribution of $\big(Y_u(a)\big)_{u\in\mathcal{U}'}$ is multivariate Gaussian. Furthermore, assume $R(y) = r_0 + r_1^\top y$ for $r_0 \in \mathbb{R}$ and $r_1 \in \mathbb{R}^J$ is an affine, real-valued function so $R(Y_u(a))$ follows a Gaussian distribution.*

A Gaussian and exchangeable model is very special and is completely defined by the mean and covariance of the joint distribution of $(Y_u(a), Y_{u'}(a))$ for any $u \neq u'$ (Aldous, 2006).

For arbitrary $u, u' \in \mathcal{U}$ with $u \neq u'$, we can write

$$\begin{bmatrix} Y_u(a) \\ Y_{u'}(a) \end{bmatrix} \;\Big|\; (Z_a = z) \sim N\left(\begin{bmatrix} \mu_{1,z} \\ \mu_{1,z} \end{bmatrix}, \begin{bmatrix} \Sigma_{1,z} + V_z & \Sigma_{1,z} \\ \Sigma_{1,z} & \Sigma_{1,z} + V_z \end{bmatrix} \right)$$

where $\mu_{1,z} \triangleq \mathbf{E}\left[Y_u(a) \mid Z_a = z\right] \in \mathbb{R}^J$ is the mean conditional on $Z_a = z$ (marginal over the latent features θ_a), $\Sigma_{1,z} + V_z \triangleq \mathrm{Cov}\left(Y_u(a)\,,\, Y_u(a) \mid Z_a = z\right) \in \mathbb{R}^{J \times J}$ is the conditional covariance matrix of a single user's engagement metrics, and $\Sigma_{1,z} \triangleq \mathrm{Cov}\left(Y_u(a)\,,\, Y_{u'}(a) \mid Z_a = z\right)$ for $u \neq u'$ is the cross-user covariance.

Lemma 2 (Multivariate Gaussian Bayesian Model)**.** *Let Assumptions 1 and 3 hold. The limit* $\theta_a \triangleq \lim_{|U| \to \infty} \frac{1}{|U|} \sum_{u \in U} Y_u(a)$ *exists almost surely. Conditional on* $\theta_a \in \mathbb{R}^J, Z_a \in \mathcal{Z}$*, the random variables* $\{Y_u(a)\}_{u \in \mathcal{U}}$ *are i.i.d. Moreover,*

$$\theta_a \mid (Z_a = z) \sim N\left(\mu_{1,z}\,,\, \Sigma_{1,z}\right) \qquad \textit{and} \qquad Y_u(a) \mid (\theta_a, Z_a = z) \sim N(\theta_a\,,\, V_z).$$

By Lemma 2 above, the latent parameter θ_a can be interpreted as the (random) almost sure limit of the empirical mean outcomes. By the above lemma, the outcome distribution P_{θ_a, Z_a} that we defined earlier is a multivariate Gaussian $N(\theta_a, V_{Z_a})$. The proof of Lemma 2 is in Appendix A.

In practice, we use historical data from previously recommended items to fit the prior distribution parameters $\mu_{1,z}, \Sigma_{1,z}$, as well as the noise covariance matrix V_z; We discuss our procedure for fitting these parameters in practice in Section 6.

4. Algorithm Overview

The *impatient bandit* algorithm we develop next builds on the classical Thompson sampling bandit algorithm (Russo et al., 2020; Thompson, 1933). However, our approach generalizes the classical Thompson sampling algorithm to take advantage of progressive feedback in settings with delayed rewards. In particular, our algorithm does this by forming a posterior for the vector θ_a, which, according to Lemma 2, defines the mean of all the engagement outcomes $(Y_u^{(1)}(a), \ldots, Y_u^{(J)}(a))$, rather than just the reward. This allows us to update the posterior distribution even after observing partial feedback. Then at decision time, we sample a random draw from the posterior of θ_a, and use that posterior sample to compute a posterior draw of the mean reward under that arm using the function R (which is linear by Assumption 3). See Algorithm 1 for more details. Note that the posterior update rule used in line 11 of Algorithm 1 can be derived using standard formulas (see further discussion in Appendix B). Additionally, we extend our algorithm to settings with user context features, where expected outcomes are linear functions of the context; See Appendix C for details.

4.1 Lower Variance Version of Algorithm

For our regret bounds, we will focus on proving results for a slightly lower-variance version of the Impatient Thompson Sampling Bandit Algorithm from Algorithm 1. We do this mainly to avoid repeating concentration of measure arguments from Qin and Russo (2023), which are quite long and complicated. We introduce this lower variance version in Algorithm 2. As m becomes large, the difference between Algorithm 1 and Algorithm 2 vanishes. Later, in

Algorithm 1 Impatient Bandit Thompson Sampling π^{TS}

Require: Priors and noise covariance matrices $\left\{\mu_{1,Z_a}, \Sigma_{1,Z_a}, V_{Z_a}\right\}_{a\in\mathcal{A}}$

1: **for** $a \in \mathcal{A}$ **do**
2: $\quad(\mu_{1,a}, \Sigma_{1,a}) \leftarrow (\mu_{1,Z_a}, \Sigma_{1,Z_a})$ ▷ Initialize beliefs.
3: $\mathcal{H}_1 \leftarrow \{Z_a : a \in \mathcal{A}\}$ ▷ Initialize history.
4: **for** $t = 1, 2, \ldots, T$ **do**
5: $\quad$**for** $u \in \mathcal{U}_t$ **do**
6: $\quad\quad\tau_u \leftarrow t$ ▷ Set user decision time.
7: $\quad\quad$**for** $a \in \mathcal{A}$ **do**
8: $\quad\quad\quad\theta'_a \sim N\left(\mu_{t,a}\,,\, \Sigma_{t,a}\right)$ ▷ Sample from belief.
9: $\quad\quad A_u \leftarrow \text{argmax}_{a\in\mathcal{A}}\left\{R(\theta'_a)\right\}$ ▷ Take action.
10: $\quad\mathcal{H}_{t+1} \leftarrow \mathcal{H}_t \cup \left\{Y_u^{(j)}(A_u) : u \in \mathcal{U}_{\le t}, j \text{ s.t. } t = \tau_u + d_j\right\}$ ▷ Update history.
11: $\quad$**for** $a \in \mathcal{A}$ **do**
12: $\quad\quad(\mu_{t+1,a}, \Sigma_{t+1,a}) \leftarrow \text{PosteriorUpdate}(\mu_{t,Z_a}, \Sigma_{t,Z_a}, V_{Z_a}, \mathcal{H}_{t+1})$ ▷ Update belief.

Section 7.2, we show empirically that Algorithm 1 behaves the same way whether m is small or large. As such, we believe that the theoretical insights we provide into the reduced-variance version carry over to the version presented in Algorithm 1.

Note that as stated in Algorithm 1, the probability that any user $u \in \mathcal{U}_t$ is assigned action a is

$$p_{t,a}^{\text{TS}} \triangleq \mathbb{P}\left(a = \underset{\alpha\in\mathcal{A}}{\text{argmax}}\, R(\tilde{\theta}_\alpha) \,\middle|\, \mathcal{H}_t\right),$$

where the probability above averages over the draw of θ'_a from the posterior $N(\mu_{t,a}, \Sigma_{t,a})$. Furthermore, lines 7–9 in Algorithm 1 can be equivalently be written as for each $u \in \mathcal{U}_t$,

$$A_u \leftarrow \left\{a \quad \text{w.p. } p_{t,a}^{\text{TS}} \text{ for all } a \in \mathcal{A}\right.$$

While typical Thompson sampling sets A_u to action a with probability $p_{t,a}^{\text{TS}}$, for each $u \in \mathcal{U}_t$, a lower variance of the algorithm can be defined which assigns exactly $|\mathcal{U}_{t,a}| \approx |\mathcal{U}_t| \cdot p_{t,a}^{\text{TS}}$ users to action a. Since $|\mathcal{U}_t| \cdot p_{t,a}^{\text{TS}}$ is not necessarily a whole number, this would involve rounding $p_{t,a}^{\text{TS}}$ to some $p_{t,a}^{\text{TS-rnd}}$ which ensures that $|\mathcal{U}_{t,a}| = |\mathcal{U}_t| \cdot p_{t,a}^{\text{TS-rnd}}$ is a whole number. This lower-variance version of the algorithm is presented in Algorithm 2.

In Algorithm 2, $\text{Round}(\left\{p_{t,a}^{\text{TS}}\right\}_{a\in\mathcal{A}}, m)$ is a function that takes the probabilities to round and the total number of samples m as input, and outputs probabilities $\left\{p_{t,a}^{\text{TS-rnd}}\right\}_{a\in\mathcal{A}}$. In order to ensure $m \cdot p_{t,a}^{\text{TS-rnd}}$ is a whole number, it rounds each $p_{t,a}^{\text{TS-rnd}}$ to a value in $\left\{0, \frac{1}{m}, \frac{2}{m}, \ldots, \frac{m-1}{m}, 1\right\}$. The rounding procedure also ensures that the rounding maintains a proper probability distribution, i.e., $1 = \sum_{a\in\mathcal{A}} p_{t,a}^{\text{TS-rnd}}$. The formal assumptions we make on the rounding procedure are written in Assumption 4.

Assumption 4 (Rounding Procedure)**.** *The rounding procedure is such that there exists a scalar* $0.5 \ge \epsilon_{\text{rnd}} > 0$ *that uniformly bounds rounding error as:*

$$\left|p_{t,a}^{\text{TS}} - p_{t,a}^{\text{TS-rnd}}\right| \le \epsilon_{\text{rnd}} \quad \textit{and} \quad p_{t,a}^{\text{TS-rnd}} / p_{t,a}^{\text{TS}} \ge 1 - \epsilon_{\text{rnd}},$$

Algorithm 2 Impatient Bandit Thompson Sampling—Lower Variance Version $\pi^{\text{TS-rnd}}$

Require: Priors and noise covariance matrices $\left\{\mu_{1,Z_a}, \Sigma_{1,Z_a}, V_{Z_a}\right\}_{a\in\mathcal{A}}$

for $a \in \mathcal{A}$ **do**
  $(\mu_{1,a}, \Sigma_{1,a}) \leftarrow (\mu_{1,Z_a}, \Sigma_{1,Z_a})$ ▷ Initialize beliefs.
$\mathcal{H}_1 \leftarrow \{Z_a : a \in \mathcal{A}\}$ ▷ Initialize history.
for $t = 1, 2, \ldots, T$ **do**
  $\left\{p^{\text{TS-rnd}}_{t,a}\right\}_{a\in\mathcal{A}} \leftarrow \text{Round}\left(\left\{p^{\text{TS}}_{t,a}\right\}_{a\in\mathcal{A}}, |\mathcal{U}_t|\right)$ ▷ Apply rounding procedure.
  $\{\mathcal{U}_{t,a}\}_{a\in\mathcal{A}} \leftarrow \text{Partition}\left(\left\{p^{\text{TS-rnd}}_{t,a}, \mathcal{U}_t\right\}_{a\in\mathcal{A}}\right)$ ▷ Note that $|\mathcal{U}_{t,a}| = |\mathcal{U}_t| \cdot p^{\text{TS-rnd}}_{t,a}$.
  for $a \in \mathcal{A}$ **do**
    for $u \in \mathcal{U}_{t,a}$ **do**
      $\tau_u \leftarrow t$ ▷ Set user decision time.
      $A_u \leftarrow a$ ▷ Take action.
  $\mathcal{H}_{t+1} \leftarrow \mathcal{H}_t \cup \left\{Y^{(j)}_u(A_u) : u \in \mathcal{U}_{\le t}, j \text{ s.t. } t = \tau_u + d_j\right\}$ ▷ Update history.
  for $a \in \mathcal{A}$ **do**
    $(\mu_{t+1,a}, \Sigma_{t+1,a}) \leftarrow \text{PosteriorUpdate}(\mu_{t,Z_a}, \Sigma_{t,Z_a}, V_{Z_a}, \mathcal{H}_{t+1})$ ▷ Update beliefs.

with probability 1 *for all* $t \in [1\colon T], a \in \mathcal{A}$.

We provide an example rounding procedure that satisfies Assumption 4 in Appendix E. In fact, this rounding procedure will ensure Assumption 4 holds for $\epsilon_{\text{rnd}} = \frac{|\mathcal{A}|}{m}$ where $m = |\mathcal{U}_1| = |\mathcal{U}_2| \cdots = |\mathcal{U}_T|$ as long as $m > 2|\mathcal{A}|$. This shows formally that rounding is not a major issue when user batches are large, as they are in the applications we have in mind.

5. Theoretical Guarantee

Nearly all bandit analyses focus on the rate at which regret—the suboptimality of selected arms—vanishes as the number of arm selections grows. When reward observations are severely delayed, there is a long delay before anything is learned, regardless of how many arm selections are made in the interim; In our motivating application, for example, the primary reward metric is computed after observing a user's 60-day engagement with a recommended item. Our theoretical focus, aligned with the practical objective, is not only on converging to the optimal arm in the limit $T \to \infty$, but also on leveraging progressive feedback to substantially improve decision quality relative to the case where no progressive feedback is available *for every* $t \in \{1, \ldots, T\}$. Two factors make this learning problem challenging:

Delayed feedback. After a user receives a recommendation, their downstream activity is revealed progressively across time. The resulting delay in information revelation limits how quickly the recommender system can learn and adapt.

Bandit feedback. The recommender system only observes a user's reward for the selected action A_u—not for all actions $\mathcal{A}$ (as is the case for full-feedback learning problems). To learn, the system needs to engage in costly exploration over the actions in $\mathcal{A}$.

We provide a regret upper bound for our Impatient Bandit algorithm that aims to reflect the above two challenges in an interpretable manner. By taking advantage of high quality

progressive feedback, it is possible for our bandit algorithm to resolve most of the uncertainty of the reward $R(Y_u)$ before one reaches the maximum delay $d_{\max} = \max_j d_j$. This progressive feedback can then be used to make higher quality decisions in this setting with bandit feedback. For our results, we prove a regret bound for the slightly lower-variance version of the Impatient Bandit Algorithm from Algorithm 2.

5.1 Warm-up: Setting without Action Features Z_a

In this section we present a regret bound for the Impatient Bandit Algorithm (Algorithm 2) in a setting in which action features Z_a are not used to tailor the prior distribution, nor the noise covariance matrix V_z (i.e., μ_z, Σ_z, V_z are constant across $z \in \mathcal{Z}$). See Section 5.2 for the generalization to the setting with action features Z_a.

5.1.1 Defining the Value of Progressive Feedback

The regret of the Impatient Bandit Algorithm fundamentally depends on the *quality* of the intermediate progressive feedback that the algorithm receives. The intermediate feedback is of higher quality if it provides more information on the true mean reward. To formalize this, below, we define a metric that captures the benefit of the progressive feedback in an information-theoretic sense. We later use this metric in our regret bound.

We define the "Value of Progressive Feedback" (VoPF) to equal the following conditional mutual information between the mean reward and the progressive (potential) outcomes, conditional on the delayed (potential) outcomes for any action $a \in \mathcal{A}$ (recall $\mathcal{A}$ is sampled according to Assumption 2):

$$\text{VoPF}(t) = I\big(\bar{R}_a;\ \underbrace{\big\{\tilde{Y}_{u,t}(a) : u \in \mathcal{U}_{<t}\big\}}_{\text{includes progressive feedback}}\ \big|\ \underbrace{\big\{\tilde{Y}_{u,t}(a) : u \in \mathcal{U}_{<(t-d_{\max})}\big\}}_{\text{only delayed outcomes}}\big). \tag{7}$$

Above, the outcomes $\big\{\tilde{Y}_{u,t}(a) : u \in \mathcal{U}_{<(t-d_{\max})}\big\} = \big\{Y_{u,t}(a) : u \in \mathcal{U}_{<(t-d_{\max})}\big\}$ refer to the *entire* vector of full-feedback outcomes, which are not censored due to a deterministic delay and are not incrementally revealed. Also note that $\big\{\tilde{Y}_{u,t}(a) : u \in \mathcal{U}_{<t}\big\}$ contains much more information about $\bar{R}_a$ than the history $\mathcal{H}_t$. Here, $\big\{\tilde{Y}_{u,t}(a) : u \in \mathcal{U}_{<t}\big\}$ represents the outcomes one would observe in a *full-feedback* setting, where the outcomes of all arms are revealed after taking an action, rather than just the selected arm's outcomes are revealed (bandit-feedback).

Note that by design, the VoPF metric is entirely a property of the underlying environment, i.e., the potential outcomes distributions of the rewards and intermediate outcomes. This means that the VoPF metric *does not* change depending on what decision-making algorithm is used to select actions.

By well-known properties of mutual information, we can rewrite the VoPF as a difference of two conditional entropies:

$$\text{VoPF}(t) = H\big(\bar{R}_a \mid \underbrace{\big\{\tilde{Y}_{u,t}(a) : u \in \mathcal{U}_{<(t-d_{\max})}\big\}}_{\text{only delayed outcomes}}\big) - H\big(\bar{R}_a \mid \underbrace{\big\{\tilde{Y}_{u,t}(a) : u \in \mathcal{U}_{<t}\big\}}_{\text{includes progressive feedback}}\big).$$

By the above, we can interpret the expected value of the VoPF as the reduction in entropy in $\bar{R}_a$ by including progressive feedback.

Remark 3 (Interpreting the Value of Progressive Feedback in Simple Examples)**.** *Consider the setting in which there are just two outcomes. For any action $a \in \mathcal{A}$, selecting $A_u = a$, the first outcome $Y_u^{(1)}(a)$ is revealed immediately ($d_1 = 0$), the second (and final) outcome $Y_u^{(2)}(a)$ is revealed with delay $d_2 = d_{\max}$, and the reward is the final outcome, $R\big(Y_u(a)\big) = Y_u^{(2)}(a)$. Assume that*

$$\Sigma_1 = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \qquad \text{and} \qquad V = \sigma_R^2 \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix},$$

for some $\rho \in [0, 1]$. Then, $\mathrm{Corr}\big(Y_u^{(1)}(a), Y_u^{(2)}(a)\big) = \rho$ and one can show that

$$\begin{aligned} \mathrm{VoPF}(t) &= \frac{1}{2} \log \left(\frac{\mathrm{Var}(\bar{R}_a \mid \{Y_u(a) : u \in \mathcal{U}_{<t-d_{\max}}\})}{\mathrm{Var}(\bar{R}_a \mid \{\tilde{Y}_{u,t}(a)\}_{u \in \mathcal{U}_{<t}})} \right) \\ &= \frac{1}{2} \log \left(\frac{\sigma_R^2 + m(t - d_{\max}) + 1}{\sigma_r^2 + m(t - d_{\max}) + 1 - \rho^2} \right) \quad \forall t \geq d_{\max}. \end{aligned}$$

If the intermediate feedback is "perfect", i.e., $\rho = 1$, then

$$\mathrm{Var}(\bar{R}_a \mid \{\tilde{Y}_{u,t}(a)\}_{u \in \mathcal{U}_{<t}}) = \mathrm{Var}(\bar{R}_a \mid \{Y_u(a)\}_{u \in \mathcal{U}_{<t}}),$$

and the Impatient Bandit algorithm will act as if the reward is observed immediately (no delay at all). Alternatively, if the intermediate feedback is completely uninformative, i.e., $\rho = 0$, then $\mathrm{VoPF}(t) = 0$. In this case, the Impatient Bandit algorithm will act as if there is no intermediate feedback at all. See Appendix D.5 for a derivation.

5.1.2 Regret Bound (Without Action Features Z_a)

Corollary 4 below follows from our main result. This corollary bounds the regret of the Impatient Bandit Thompson Sampling algorithm ($\pi^{\text{TS-rnd}}$ from Algorithm 2) in any batch in terms of (i) the reward variance $\sigma_R^2 \triangleq \mathrm{Var}\big(R(Y_u(a)) \mid \bar{R}_a\big)$, (ii) the total number of previous user interactions over the number of actions $|\mathcal{U}_{<t}|/|\mathcal{A}|$, (iii) the Value of Progressive Feedback, and (iv) rounding error due to implementing the Thompson sampling allocation in a finite batch.

Recall that $|\mathcal{U}_{<t}| = m(t-1)$ where $m = |\mathcal{U}_1| = |\mathcal{U}_2| = \cdots = |\mathcal{U}_T|$ is the user batch size. Summing across t gives a cumulative regret bound, which is more conventional. We prefer this, stronger, bound which indicates how low regret is in every batch.

Corollary 4 (Regret of Impatient Bandit Algorithm without Action Features Z_a)**.** *Let Assumptions 1-4 hold, and let Σ_1 and V be invertible. Then under the Impatient Bandit Thompson Sampling algorithm $\pi^{\text{TS-rnd}}$ (Algorithm 2), for any $t > 1$,*

$$\mathbf{E}\left[\Delta_t(\pi^{\text{TS-rnd}})\right] \leq \underbrace{\exp\big[-\mathrm{VoPF}(t)\big]}_{\substack{\text{Decrease from} \\ \text{Progressive Feedback}}} \cdot \underbrace{\sigma_R \sqrt{\frac{2|\mathcal{A}| \log |\mathcal{A}|}{\sigma_R^2 (r_1^\top \Sigma_1 r_1)^{-1} + |\mathcal{U}_{<(t-d_{\max})}|}}}_{\text{Regret under Delayed Rewards}} + \underbrace{O(\epsilon_{\text{rnd}})}_{\text{Rounding error}} . \tag{8}$$

We discuss each term in this bound:

- **Regret under Delayed Rewards:** Above, we can interpret the term $\sigma_R \cdot \sqrt{\frac{2|\mathcal{A}|\log(|\mathcal{A}|)}{\sigma_R^2(r_1^\top \Sigma_1 r_1)^{-1}+|\mathcal{U}_{<(t-d_{\max})}|}}$ as the regret the Thompson Sampling algorithm would achieve with delayed rewards. In this setting, the system has observed rewards associated with recommendations that were made more than $d_{\max}$ periods ago, amounting to $|\mathcal{U}_{<(t-d_{\max})}|$ observed interactions in total. The system has observed no feedback on more recent recommendations. This term can also be interpreted as a bound on the regret of an algorithm that *ignores incremental feedback* even though it is available.

  The factor of the number of actions $|\mathcal{A}|$ in the bound can be thought of as the price of bandit feedback. It appears in our analysis because the system only observes a user's interaction *with the single item recommended to them.* In a model in which we saw each user's interaction with all items in $\mathcal{A}$, this term would be replaced with 1 in our bound. The $\log(|\mathcal{A}|)$ term appears because it is used as a coarse upper bound for the entropy of the optimal action A^* given the history.

- **Decrease in Regret due to Progressive Feedback:** When using progressive feedback, the regret upper bound gets rescaled by a function of the of value of progressive feedback, $\exp\big[-\text{VoPF}(t)\big]$. Note in (8) above that when $\text{VoPF}(t) = 0$, i.e., the intermediate outcomes are completely uninformative, the regret bound equals the standard regret bound for Thompson Sampling one would expect without any progressive feedback and just delayed rewards.

- **Rounding error:** Our analysis makes the $O(\epsilon_{\text{rnd}})$ term explicit. We hide it in a big-O because we do not consider it to be an interesting feature of this analysis. In our application, the batch size $m = |\mathcal{U}_t|$ is large (hence ϵ_{rnd} is not a big concern), but so is σ_R, so many batches are still required.

Per-User Regret Across Batches In Extreme Cases. In Proposition 5 below, we explicitly bound the average per-user regret across batches $\mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T}\Delta_t(\pi^{\text{TS-rnd}})\right]$ under extreme values of $\text{VoPF}(t)$. At one extreme, we consider the case in which the value of progressive feedback is zero (intermediate outcomes are completely uninformative). Regardless of the number of users per-batch m, regret can only be small after waiting to observe delayed rewards ($T > d_{\max}$); This takes 60 days in our main motivating application. At the other extreme, we consider the case in which there are "perfect" intermediate outcomes (as if there is no delay). We can interpret Corollary 4 as interpolating between these two extreme cases when $\text{VoPF}(t)$ takes on different values.

Proposition 5 (Regret Under Extremes)**.** *Let the conditions of Corollary 4 hold.*
(1) Completely Uninformative Intermediate Feedback (Delayed Rewards). *In the case that the intermediate feedback is completely uninformative, i.e.,* $\text{VoPF}(t) = 0$ *for all* t*,*

$$\mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T}\Delta_t(\pi^{\text{TS-rnd}})\right] \leq \frac{d_{\max}}{T}\sqrt{2(r_1^\top \Sigma_1 r_1)\log(|\mathcal{A}|)} + 3\sigma_R\sqrt{\frac{2|\mathcal{A}|\log(|\mathcal{A}|)}{Tm}} + O(\epsilon_{\text{rnd}}). \quad (9)$$

(2) Perfect Intermediate Feedback (No Delay). *If we have a perfect surrogate, i.e.,* $\mathrm{Corr}\big(Y_u^{(1)}, R(Y_u)\big) = 1$ *and* $d_1 = 0$,

$$\mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T}\Delta_t(\pi^{\text{TS-rnd}})\right] \leq 3\sigma_R\sqrt{\frac{2|\mathcal{A}|\log(|\mathcal{A}|)}{Tm}} + O(\epsilon_{\text{rnd}}). \tag{10}$$

The regret bound in (10) matches (excluding constant factors) those for standard Thompson Sampling, without delayed rewards, from the literature (Russo and Van Roy, 2016). Note, the gap in regret between the two extreme settings above is worsened when one considers longer-term outcomes, i.e., as the delay $d_{\max}$ grows.

Novelty of our Regret Bound. Our regret bound is novel because of how it is controlled by the Value of Progressive Feedback (VoPF). The bound directly quantifies the expected benefit of incorporating progressive feedback over just using delayed rewards. It is notable that the regret bound applies to settings in which rewards are completely delayed (intermediate outcomes have no useful information and the VoPF is zero), settings in which there is no delay (intermediate outcomes are perfect), as well as settings in between these two extremes. Moreover, in practice, the VoPF is a useful metric that can be estimated from historical data and could be used to compare the quality of different progressive feedback signals. In Section 6.1, we demonstrate how we estimate the VoPF on a Spotify podcast recommendation dataset.

To provide a regret bound in terms of the Value of Progressive Feedback metric, we had to overcome substantial conceptual and mathematical subtleties. The challenge lies in the nature of VoPF(t). This metric measures the value of early feedback on an arm that *could be* available at decision-time t, but only *if the algorithm had chosen to repeatedly recommend that arm in earlier periods.* When the VoPF(t) is low, progressive feedback would not be especially useful even if the algorithm opted to explore aggressively. But to upper bound the regret incurred at the t^{th} decision time, we need to reason about the volume of information the algorithm *did* acquire (i.e. its information gain) rather than the volume of information it *could have* acquired (the VoPF). The mismatch between these notions is severe, since our Thompson sampling based algorithm will choose to select some arms very rarely. That is, it often gathers far less information about some arms than the VoPF measure suggests. At a high level, there is hope of overcoming this mismatch because Thompson sampling selects arms infrequently only when they are unlikely to be optimal and hence do not matter much to decision-making.

To formalize this, we deviate from conventional analyses in the literature and provide intricate proofs that use inverse-propensity weights to analyze the evolution of posterior beliefs, building on Qin and Russo (2023). It is unclear if there is any hope of adapting more conventional analyses, like the use confidence interval-based analyses (Russo and Van Roy, 2014; Lattimore and Szepesvári, 2020) and those that utilize the *information ratio* (Russo and Van Roy, 2016; Lattimore and Szepesvári, 2019, 2020), which were not designed to cope with delayed feedback.

Overview of Proof Approach. We now provide a brief overview of the key steps used in the proof of our regret bound from Corollary 4. Let $\sigma_{t,a}^2 \triangleq \mathrm{Var}(\bar{R}_a \mid \mathcal{H}_t)$, where recall from (1) that $\bar{R}_a \triangleq \mathbf{E}\left[R\big(Y_u(a)\big) \mid \theta_a\right]$. By Lemma 7, which uses the probability matching

property of Thompson Sampling and rounding Assumption 4, we have that

$$\mathbf{E}\left[\Delta_t(\pi^{\text{TS-rnd}})\right] \leq \sqrt{2\log(|\mathcal{A}|)\mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\sum_{a\in\mathcal{A}} p^*_{t,a}\sigma^2_{t,a}\right]} + O(\epsilon_{\text{rnd}})$$

where $p^*_{t,a} = \mathbb{P}(A^* = a \mid \mathcal{H}_t)$. Thus, the instantaneous regret is controlled by a function of the posterior uncertainty of the optimal arm under the Thompson sampling algorithm.

The crux of the proof is showing the following inequality holds:

$$\mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\sum_{a\in\mathcal{A}} p^*_{t,a}\sigma^2_{t,a}\right] \leq (1+2\epsilon_{\text{rnd}})\sum_{a\in\mathcal{A}}\mathbf{E}\left[\text{Var}\left(\bar{R}_a \mid Z_a, \{\tilde{Y}_{u,t}(a)\}_{u\in\mathcal{U}_{<t}}\right)\right]$$

Crucially above, the left hand side is an expectation related to the actions selected under the rounded Impatient Bandit algorithm, but the right hand side is a function of the potential outcomes and not the algorithm itself. The proof of this inequality uses inverse-propensity weighting together with an inequality from Qin and Russo (2023).

Finally, the corollary holds by Lemma 11, which shows the following equality

$$\mathbf{E}\left[\text{Var}\left(\bar{R}_a \mid Z_a, \{\tilde{Y}_{u,t}(a)\}_{u\in\mathcal{U}_{<t}}\right)\right] = \mathbf{E}\left[\frac{\sigma^2_R(Z_a)\cdot\exp\big(-2\cdot\text{VoPF}(t,Z_a)\big)}{\sigma^2_R(Z_a)\cdot(r_1^\top\Sigma_{Z_a}r_1)^{-1} + |\mathcal{U}_{<(t-d_{\max})}|}\right].$$

5.2 General Setting with Action Features Z_a

We now characterize the regret of the Impatient Bandit Thompson Sampling algorithm that uses action features Z_a. First, we define a generalized version of the Value of Progressive Feedback that depends on Z_a:

$$\text{VoPF}(t,z) \triangleq I\Bigg(\bar{R}_a;\ \underbrace{\{\tilde{Y}_{u,t}(a) : u\in\mathcal{U}_{<t}\}}_{\text{includes progressive feedback}}\ \mid \underbrace{\{\tilde{Y}_{u,t}(a) : u\in\mathcal{U}_{<(t-d_{\max})}\}}_{\text{only delayed outcomes}}, Z_a = z\Bigg). \quad (11)$$

Using this generalized version of the Value of Progressive Feedback, we can now present our generalized regret bound. This result bounds the regret of Impatient Bandit Thompson sampling in terms of (i) the variance of rewards conditional on action features $\sigma^2_R(z) = \text{Var}\big(R(Y_u(a)) \mid \bar{R}_a, Z_a = z\big)$, (ii) the number of previous recommendations $|\mathcal{U}_{<t}|$, (iii) the number of actions/items $|\mathcal{A}|$, (iv) the generalized definition of the value of progressive feedback (VoPF), and (v) a rounding error term.

Theorem 1 (Regret with Progressive Feedback)**.** *Let Assumptions 1-4 hold, and let $\Sigma_{1,z}$ and V_z be invertible for all $z\in\mathcal{Z}$. Then under Impatient Bandit Thompson Sampling algorithm $\pi^{\text{TS-rnd}}$ (Algorithm 2), for any $t>1$,*

$$\mathbf{E}\left[\Delta_t(\pi^{\text{TS-rnd}})\right] \leq \sqrt{\mathbf{E}\left[\frac{\exp\big(-2\cdot\text{VoPF}(t,Z_a)\big)\cdot\sigma^2_R(Z_a)}{\sigma^2_R(Z_a)\cdot(r_1^\top\Sigma_{Z_a}r_1)^{-1} + |\mathcal{U}_{<(t-d_{\max})}|}\right]} \times \sqrt{2|\mathcal{A}|\log(|\mathcal{A}|)} + O(\epsilon_{\text{rnd}}).$$

Note that the expectation $\mathbf{E}\left[\frac{\exp(-2\cdot\text{VoPF}(t,Z_a))\cdot\sigma^2_R(Z_a)}{\sigma^2_R(Z_a)\cdot(r_1^\top\Sigma_{Z_a}r_1)^{-1}+|\mathcal{U}_{<(t-d_{\max})}|}\right]$ above averages over the draw of action features Z_a.

6. Fitting the Prior

We now discuss how to use historical data (training set), collected prior to the deployment of the Impatient Bandit algorithm, to fit the prior distribution parameters $\{\mu_{1,z}, \Sigma_{1,z}\}_{z\in\mathcal{Z}}$ and noise covariance matrices $\{V_z\}_{z\in\mathcal{Z}}$. We use $\mathcal{U}^{\text{hist}}$ to denote the historical users (training set) and $\mathcal{A}^{\text{hist}}$ to denote the historical items which were recommended to users $\mathcal{U}^{\text{hist}}$. We assume items $\mathcal{A}^{\text{hist}}$ are drawn according to the process from Assumption 2. Throughout, we assume that there are a finite number of action features, i.e., $|\mathcal{Z}| < \infty$. In the podcast recommendation case, Z_a could be the category of the podcast (e.g., comedy, news).

Fitting the Noise Covariance Matrices. Recall that $Y_u(a) \mid (\theta_a, Z_a = z) \sim N(\theta_a, V_z)$. We estimate the V_z by averaging the noise covariance matrices $\hat{V}^{(a)}$ estimates for each action $a \in \mathcal{A}^{\text{hist}}$:

$$\hat{V}_z \triangleq \frac{\sum_{a\in\mathcal{A}^{\text{hist}}} \mathbb{1}_{Z_a=z}\hat{V}^{(a)}}{\sum_{a\in\mathcal{A}^{\text{hist}}} \mathbb{1}_{Z_a=z}} \quad \text{where} \quad \hat{V}^{(a)} \triangleq \frac{1}{N^{(a)}} \sum_{u\in\mathcal{U}^{\text{hist}}} \mathbb{1}_{A_u=a}\big(Y_u - \bar{Y}^{(a)}\big)\big(Y_u - \bar{Y}^{(a)}\big)^\top.$$

Above, we use, $N^{(a)} \triangleq \sum_{u\in\mathcal{U}^{\text{hist}}} \mathbb{1}_{A_u=a}$ and $\bar{Y}^{(a)} \triangleq \frac{1}{N^{(a)}} \sum_{u\in\mathcal{U}^{\text{hist}}} \mathbb{1}_{A_u=a} Y_u$.

Fitting the Prior Distribution Parameters. In the statistics and machine learning literature, a common approach to fitting an informative prior distribution using data is type II maximum likelihood (Murphy, 2022, page 173)—also called empirical Bayes (Casella, 1985). For the Impatient Bandit algorithm, the prior hyperparameters we want to fit using historical data are $\{\mu_{1,z}, \Sigma_{1,z}\}_{z\in\mathcal{Z}}$, i.e., the mean vectors and covariance matrices for the multivariate Gaussian prior for $\theta_a \mid Z_a$. For now, we assume that we already have estimators of the noise covariance matrices $\{V_z\}_{z\in\mathcal{Z}}$.

The basic idea behind type II maximum likelihood is to choose the prior hyperparameters $\{\mu_{1,z}, \Sigma_{1,z}\}$ that maximize the *marginal* likelihood of the data. In our Gaussian prior case, the maximum marginal likelihood criterion can be written as the following minimization problem:

$$(\hat{\mu}_{1,z}, \hat{\Sigma}_{1,z}) \triangleq \operatorname{argmin}_{\mu,\Sigma} \sum_{a\in\mathcal{A}^{\text{hist}}} \mathbb{1}_{Z_a=z}\Bigg\{ \log\left(\Sigma + V_z/N^{(a)}\right) + \underbrace{(\bar{Y}^{(a)} - \mu)^\top \left[\Sigma + V_z/N^{(a)}\right]^{-1} (\bar{Y}^{(a)} - \mu)}_{\text{Weighted Least Squares}} \Bigg\}$$

Note that by the above criterion, it is clear that if $\Sigma_{1,z}$ was known, then $\hat{\mu}_{1,z}$ would be the solution to a weighted least squares criterion. In general, when $\Sigma_{1,z}$ is unknown, one can minimize the above criterion using Newton-Raphson or Expectation maximization (Normand, 1999; SAS, 2018).

In the special case that $N^{(a)} = n$ for all $a \in \mathcal{A}^{\text{hist}}$ such that $Z_a = z$, then the solution to the above $\hat{\mu}_{1,z}$ is a simple mean over $\bar{Y}^{(a)}$'s (no matter the value of $\Sigma_{1,z}$):

$$\hat{\mu}_{1,z} = \frac{1}{\sum_{a\in\mathcal{A}^{\text{hist}}} \mathbb{1}_{Z_a=z}} \sum_{a\in\mathcal{A}^{\text{hist}}} \mathbb{1}_{Z_a=z}\bar{Y}^{(a)}. \tag{12}$$

For our experiments, we approximately solve by estimating $\mu_{1,z}$ using (12) (even though $N^{(a)}$ varies across actions). We estimate $\hat{\Sigma}_{1,z}$ using $\hat{\Sigma}_{1,z} \triangleq \frac{1}{\sum_{a\in\mathcal{A}^{\text{hist}}} \mathbb{1}_{Z_a=z}} \sum_{a\in\mathcal{A}^{\text{hist}}} \mathbb{1}_{Z_a=z}\big(\bar{Y}^{(a)} - \hat{\mu}_{1,z}\big)\big(\bar{Y}^{(a)} - \hat{\mu}_{1,z}\big)^\top$.

6.1 Example on Real-World User Interaction Data

To illustrate prior fitting in practice, we consider a dataset of podcast consumption traces collected on the Spotify audio streaming platform between September 2021 and May 2022. The data is divided into a training set and an independent validation set; The training and validation sets cover podcast shows appearing during the periods September–December 2021 and January–March 2022, respectively. Each subset consists of a sample of 200 podcast shows first published on the platform during a given three-month period. For each of these shows, the data contains a representative sample of users that discover the show during the same three-month period. For each user, we obtain a longitudinal trace that captures their engagement with the show on each day starting from the discovery, i.e.,

$$Y_u^{(j)} = \mathbb{1}\{\text{user } u \text{ engaged with the show on the } j\text{th day since discovery}\},$$

for $j = 1, \ldots, 60$. By construction, the user engaged with (discovered) the recommended content on the first day, i.e., $Y_u^{(1)} = 1$ for all u. The reward is defined as the cumulative engagement over the 60 days from the initial discovery, i.e.,

$$R(Y_u) = \sum_{j=1}^{60} Y_u^{(j)}$$

In total, the dataset consists of 8.77M activity traces, corresponding to a total of 26M cumulative active-days. The number of traces per show ranges between 2.4K and 295K, with a median of 5.8K.

We examine the prior fitted using the procedure described above using the data from training set. First, in Figure 3 (left), we examine the predictive error of the model on the validation set as we vary the number of days and the batch size. Specifically, we compute

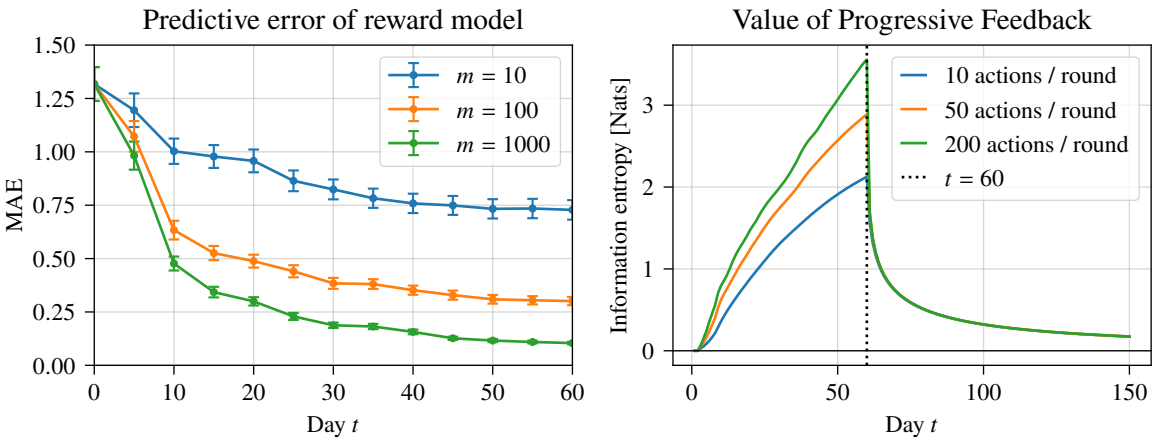

Figure 3: Prior learned on Spotify interaction data. Left: predictive error of the reward model as a function of the number of traces $m \in \{10, 100, 1000\}$ and the number of observed days t. The mean and standard errors are computed over 200 actions in the test set. Right: estimated Value of Progressive Feedback for $m \in \{10, 50, 200\}$.

the mean absolute error of the observed rewards and posterior mean of our model, which is initialized with the fitted prior. We see that the model's predictions can be relatively accurate after observing only 10 days of data. The predictions improve as time passes, and increasing the batch size further increases predictive accuracy.

In Figure 3 (right), we examine the Value of Progressive Feedback according to the fitted prior. We plot the Value of Progressive Feedback as we vary the batch size m. Notice that as m increases, the value of surrogate feedback increases, as the larger batch sizes reduce noise.

7. Empirical Evaluation

Next, we investigate different aspects of the performance of our bandit algorithm empirically with two decision-making problems. After briefly introducing these two problems in Section 7.1, we address five questions.

1. How does the empirical improvement from progressive feedback compare to our theory's predictions? (Section 7.2)

2. Are there real-world applications that can benefit substantially from progressive feedback? (Section 7.3)

3. How important is fitting the prior, as described in Section 6? (Section 7.4)

4. How do bandits with and without progressive feedback fare in a realistic setting with perpetually changing actions? (Section 7.5)

5. How does Thompson sampling compare to sequential elimination in problems with progressive feedback? (Section 7.6)

To this end, we quantify empirically the performance of our impatient bandit algorithm on the two problems. A natural benchmark for a bandit algorithm that makes use of progressive feedback is to compare it to a copy of the same algorithm, but one where we pretend that all outcomes are revealed at once after delay $d_{\max}$. Equivalently, we can think of that benchmark as ignoring the outcomes $Y_u^{(1)}, Y_u^{(2)}, \ldots$ as they become available, and only using the final delayed reward $R(Y_u)$. This benchmark is implicit in Theorem 1, and will appear throughout our experiments.

7.1 Data-Generating Environments

Across our experiments, we study two decision-making problems. The first is an idealized, synthetic problem, where we can explicitly vary the Value of Progressive Feedback. The second one is inspired by a real-world product problem at Spotify (a problem we will later re-examine in Section 8). In this second problem, actions, rewards and progressive feedback represent actual interactions of users with content on Spotify.

7.1.1 Synthetic Environment

We postulate a simple, synthetic generative model. There are $J = 25$ outcomes per decision denoted $Y_u \in \mathbb{R}^J$, and each outcome $j \in [J]$ is revealed with delay $d_j = j - 1$ (first outcome

is revealed immediately, second outcome is revealed following decision time $t+1$, and so on). Given a hyperparameter $\alpha \in [0, 1)$, for each action $a \in \mathcal{A}$, we sample a mean outcome vector

$$\theta_a \sim N(0, \Sigma_1), \quad \text{where} \quad \Sigma_1 = (1-\alpha^2)\boldsymbol{I} + \alpha^2 \mathbf{1}\mathbf{1}^\top, \tag{13}$$

where $\boldsymbol{I}$ is the $J \times J$ identity matrix, and $\mathbf{1}$ is the all-ones vector. Then, conditional on θ_a for each action a and each user u, we independently sample potential outcomes

$$Y_u(a) \mid \theta_a \sim N(\theta_a, V), \quad \text{where} \quad V = m\boldsymbol{I}.$$

Note that the dependency of the noise covariance matrix V on the batch size m is chosen such that, informally, the amount of information collected at each time step is independent of m. Furthermore, α controls how much information the first entries of θ_a contain about future entries. The higher α is, the more informative the progressive feedback is. In our experiments, we vary α to vary how informative the intermediate leading indicators are. Specifically, we vary $\alpha \in \{0.1, 0.4, 0.8\}$. We refer to these as the *Low*, *Mid*, and *High* information regimes, respectively.

We consider a reward that is the average outcome: $R(Y_u) = c^{-1} \sum_{j=1}^{J} Y_u^{(j)}$, revealed after delay $J-1$. We set $c = \sqrt{(1-\alpha^2)J + \alpha^2 J^2}$ such that the mean reward $\bar{R}_a = \mathbf{E}[R(Y_u(a)) \mid \theta_a] = c^{-1} \sum_{j=1}^{J} \theta_a^{(j)}$ has variance $\mathrm{Var}[\bar{R}_a] = 1$ for any choice of parameters.

7.1.2 Spotify Environment

We reuse the data presented in Section 6.1, consisting of discoveries of podcast shows on Spotify, as well as the user's subsequent engagement with the show over the 60 days that follow. In this semisynthetic environment, actions correspond to podcast shows. Every

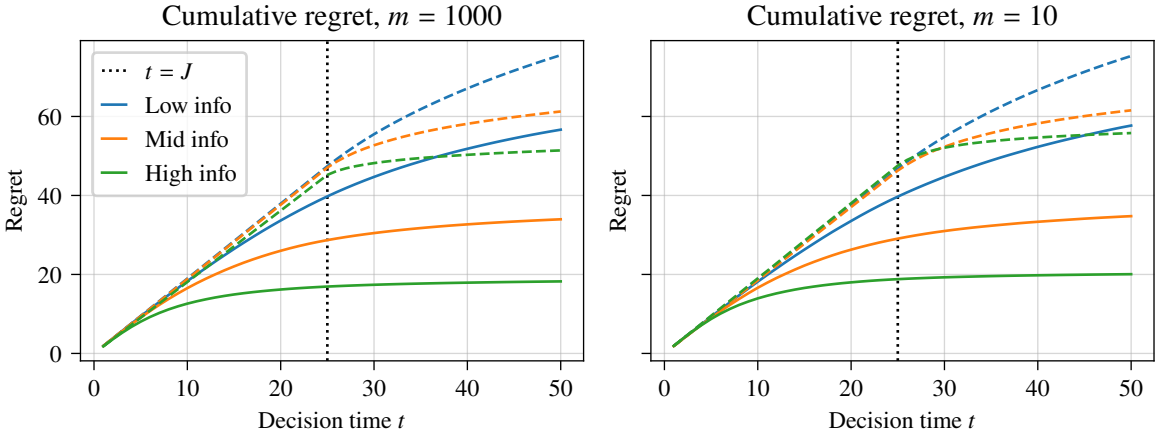

Figure 4: Cumulative regret for three different settings of the generative model, averaged over 500 runs. Solid lines correspond to the Impatient Bandit algorithm, which uses progressive feedback, and dashed lines correspond to a Thompson Sampling algorithm that ignores progressive feedback and waits for delayed rewards. All standard errors are ≤ 0.88.

time we take an action, we sample a datum Y_u from the user engagement traces of the corresponding show uniformly at random with replacement. The reward is defined as the cumulative engagement: $R(Y_u) = \sum_{j=1}^{J} Y_u^{(j)}$.

7.2 How does the empirical improvement from progressive feedback compare to our theory's predictions?

We run simulations on the synthetic environment with $|\mathcal{A}| = 20$ actions. In Figure 4 we plot the cumulative regret of the Impatient Bandit algorithm in the low, medium, and high information regimes for $m \in \{10, 1000\}$. We contrast the performance of our Impatient Bandit algorithm, Algorithm 1 (solid lines), with that of a Thompson Sampling algorithm that only uses full rewards observed after delay $d_{\max} = J - 1$ (dashed lines). Note that both the Impatient Bandit and delayed Thompson Sampling algorithms use a correctly specified prior that is derived using (13).

We observe that the benefit of using intermediate outcomes increases with α. Furthermore, the average performance of the algorithms does not change substantially as we reduce the number of actions per round dramatically from $m = 1000$ to $m = 10$, suggesting that the sampling step does not impact the performance in expectation.

In Figure 5, on the left we display the Value of Progressive Feedback (11) for the low, mid, and high information regimes. Recall that the VoPF is a property of the underlying data generating environment, and can be computed in closed form given our generative model. On the right-hand side of Figure 5 we plot the empirical ratio (on log scale) between the regret of Thompson Sampling using only delayed rewards and the regret of the Impatient Bandit algorithm, which uses progressive feedback. Based on Corollary 4, we expect that the logarithm of this regret ratio is approximately upper bounded by the Value of Progressive Feedback. Indeed, we observe that the (log) empirical regret ratio on the right closely

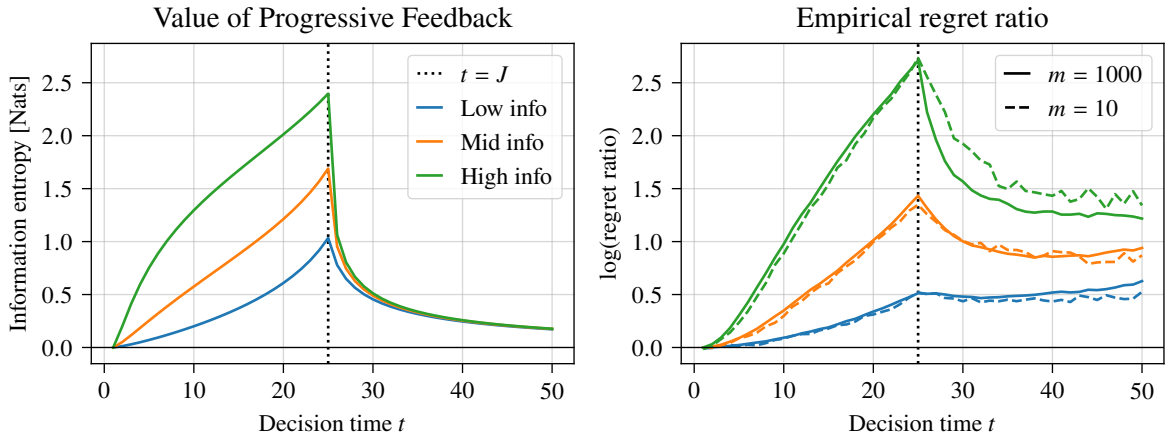

Figure 5: A comparison between the Value of Progressive Feedback, a theoretical quantity (*left)* and the empirical benefit of our algorithm over the delayed bandit algorithm, measured by averaging over 500 simulations (*right*, experimental details provided in the text). The empirical benefit qualitatively matches the theoretical prediction.

resemble the VoPF(t) curves on the left. Again, we observe no significant difference between the large-batch ($m = 1000$) and small-batch ($m = 10$) regimes.

Notice that the VoPF sharply drops at decision time $t = J + 1 = 26$. This occurs because, when $t = J + 1$, the set of delayed outcomes $\left\{Y_u(a) : u \in \mathcal{U}_{<(t-J)}\right\}$ begins to include the user potential outcomes from the first batch.

7.3 Are there real-world applications that can benefit substantially from progressive feedback?

Next, we focus on the Spotify environment, where intermediate outcomes and rewards are collected from real-world interactions, and reflect a concrete product problem. This setting was first described in Section 6.1. We ask: Does the value of progressive feedback show a large benefit on real-world data?

We benchmark our Impatient Bandit algorithm (which we label *Progressive* in the following) against several baselines, all of which are based on variants of Thompson sampling. In addition to the *Delayed* algorithm we introduced previously, we consider two additional algorithms.

Day-two proxy. Instead of the true delayed reward, this algorithm learns to maximize a short-term proxy. Specifically, the algorithm uses $Y_u^{(2)}$ as a proxy for the reward, and discards all subsequent information $(Y_u^{(j)})_{j \geq 3}$. This baseline captures an intuitive outcome that is clearly related to the goal of maximizing habitual engagement: Does the user return to the show the day after first discovering it? This baseline is representative of short-term proxies widely used in recommender systems, such as the click-through-rate, the dwell time, or the conversion rate (Lalmas et al., 2014; Dedieu et al., 2018; Bogina and Kuflik, 2017).

Oracle. This algorithm assumes that all outcomes are observed immediately following an action, i.e., that $d_j = 0$ for all j. This is clearly unrealistic, but it provides a natural upper-bound on attainable performance.

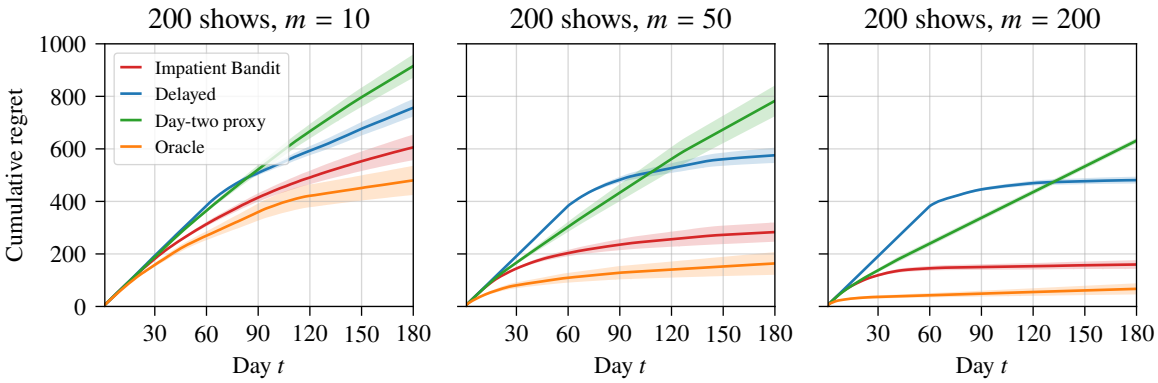

Figure 6: Cumulative regret of different bandit algorithms on the Spotify problem with $|\mathcal{A}| = 200$ shows and batch sizes $m \in \{10, 50, 200\}$. The mean (solid line) and standard error (shaded area) are computed over 10 runs.

Note, all the algorithms use the same prior learned via the procedure described in Section 6.

We consider a setting with $|\mathcal{A}| = 200$ and we vary the size of the batches (m). We run 10 simulations for each algorithm, and we plot the resulting (average) cumulative regret in Figure 6. These regret plots suggest that using the day-two outcome is a poor proxy for the true reward, since the Day-two proxy algorithm appears to be the only one that does not have sublinear regret. The performance of the Delayed algorithm is comparable to that of the Day-two proxy until the day 60, at which the algorithm begins to observe the delayed rewards. The Progressive algorithm accumulates much less regret during the first 60 decision times compared to the Delayed algorithm. Strikingly, the Progressive algorithm's regret appears to be only slightly worse than that of the oracle.

7.4 How important is fitting the prior, as described in Section 6?

We now examine the importance of using a learned prior, in the context of the Spotify decision-making problem. To this end, we include an additional baseline, which we call *Progressive (isotropic)*. Specifically, this variant uses the Impatient Bandit algorithm, but instead of using the fitted prior it uses the following uninformative prior:

- the prior mean is the all-zeros vector: $\mu_1 = \mathbf{0}$,
- the prior covariance is isotropic (multiple of identity matrix): $\Sigma_1 = 100 \cdot I$, and
- the noise covariance matrix is the identity matrix: $V = I$.

In Figure 7 we see that the cumulative regret of the progressive algorithm with the uninformative prior is comparable to that of the Delayed algorithm. Both the Progressive (uninformative prior) and Delayed algorithms are significantly outperformed by the Progressive algorithm with the fitted prior. These results suggest that the prior is really *driving* the good performance of the Impatient Bandit algorithm.

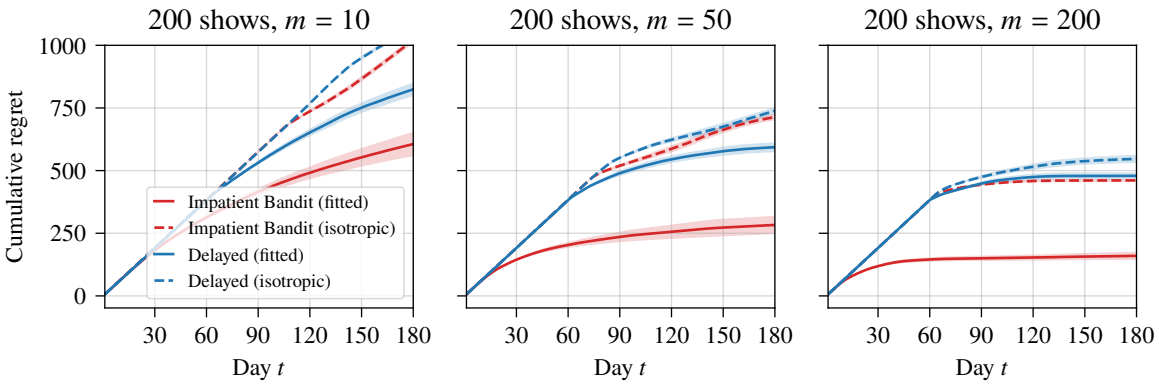

Figure 7: Cumulative regret of two variants of our bandit algorithm using different prior and covariance matrices, and a delayed bandit algorithm. The mean and standard error are computed over 10 runs. Accurately estimating hyperparameters is instrumental to achieving good performance.

7.5 How do progressive feedback bandits fare in a realistic setting with perpetually changing actions?

We consider a non-stationary variant of the Spotify environment, where the set of shows which the algorithm recommends from gradually changes at each decision time. This is designed to reflect how in the real podcast recommendation problem, new podcast shows are continually being released over time. Specifically, in our experiment, the number of available actions is equal to 60 throughout. After each decision-time, we drop the oldest action, and add a new action.

In Figure 8 we plot the cumulative regret of our algorithm over time up until $t = 340$ days. Note in this setting there are 400 shows in the action space in total over time; this means that each action is selected less than once on average. In this setting, it is particularly striking that the regret of the Delayed algorithm matches or is slightly worse than that of the Day-two proxy algorithm. Since the set of shows to recommend is changing, the Delayed algorithm's regret does not begin to improve at day 60, as we saw earlier in Figure 6. These results suggest that in settings where the cold start problem is at the forefront (e.g., a recommendation problem where the set of items over which to recommend changes frequently (Baran et al., 2023)), the Delayed algorithm can perform very poorly. In contrast, the algorithm that incorporates progressive feedback incurs less than half of the cumulative regret of the Delayed and Day-two proxy algorithms.

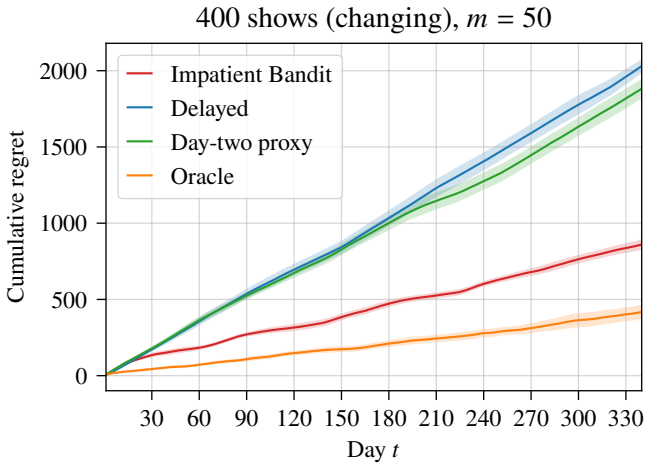

Figure 8: Cumulative regret of different bandit algorithms on a variant of the Spotify problem, where the action set changes slightly at each decision time. The mean (solid line) and standard error (shaded area) are computed over 10 runs.

7.6 How does Thompson sampling compare to sequential elimination?

In this section, we compare our algorithm to a Sequential Elimination based approach (Even-Dar et al., 2006; Shahrampour et al., 2017; Karnin et al., 2013). Sequential elimination is a natural algorithm to compare to since it is conceptually very similar to Thompson Sampling, and it can also take advantage of the Bayesian filter we introduce. The sequential elimination algorithm we use maintains a posterior distribution over the mean reward using

our progressive feedback Bayesian model. Under Sequential Elimination, when an action's probability of being optimal according to the posterior distribution falls below a certain threshold, that action is eliminated; Actions are selected uniformly over the set of non-eliminated actions. In contrast, Thompson Sampling gradually reduces the probability of selecting arms that are unlikely to be optimal, which makes it often more difficult to analyze.

For the purposes of this comparison, we use the synthetic environment and focus on the low and mid-information regimes. We run simulations with two variants of Sequential Elimination, with threshold values of 1% and 4%, respectively. We plot the cumulative regret of both successive elimination algorithms and our Impatient Bandit Thompson Sampling based algorithm in Figure 9. Similar to our Impatient Bandit algorithm, the regret of the successive elimination algorithms improve in the mid-information environment, as compared to the low information environment. We find that the Impatient Bandit Thompson Sampling based algorithm matches the performance of the best successive elimination algorithm we consider in all settings, and has even better performance in the mid-information regime. Overall, while successive elimination may have some advantages, it does not quite get the same performance as Thompson Sampling.

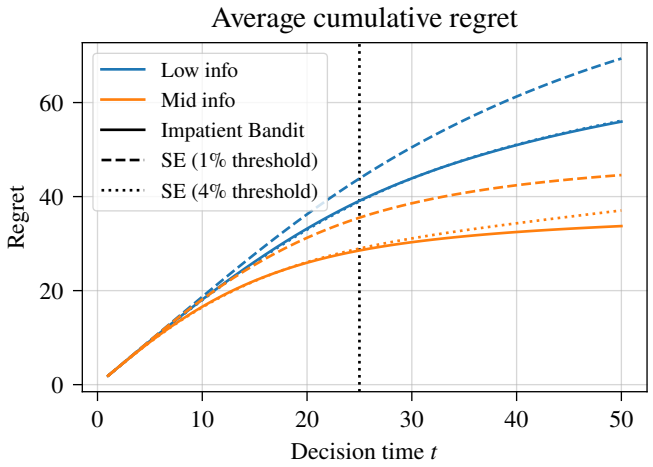

Figure 9: Cumulative regret of our Thompson Sampling Impatient Bandit algorithm (solid lines) and two variants of Sequential Elimination with different elimination thresholds (dashed and dotted lines), averaged over 5000 runs. All standard errors are ≤ 0.28. The dashed vertical line indicates the value of $d_{\max}$, the delay before any reward is fully observed.

8. A/B Test

Our work addresses a key challenge encountered at Spotify. While Maystre et al. (2023) demonstrated substantial improvements from optimizing recommender systems for long-term outcomes, their methods introduce significant delays in evaluating new content, as they require waiting to observe whether users form lasting listening habits with the content. This section describes how progressive feedback, combined with the Gaussian filtering procedure introduced in Sections 3 and 4, alleviates this challenge. It also presents evidence of significant

impact from an A/B test involving hundreds of millions of users over a two-month period. Based on this and subsequent tests, the method was rolled out in production to power personalized podcast recommendations for hundreds of millions of listeners worldwide.

While our model of progressive feedback in multi-armed bandits aims to distill a core challenge that arises in recommender systems, rather than attempt to realistically model all intricacies of a particular system, the test results demonstrate that incorporating progressive feedback via Gaussian filtering can have enormous impact even in a complex industrial-scale system. The primary differences between the algorithm introduced in previous sections and the A/B test implementation are that *a*) the Gaussian filtering model is used to rank items on a shelf rather than make single recommendations, and *b*) the algorithm tested does not perform purposeful exploration, which was not a priority for our test partners. These differences are discussed in Section 8.4. Despite these differences, the substantial improvements observed in the A/B test validate the fundamental insight that progressive feedback can be leveraged to accelerate learning in recommendation systems.

8.1 Recommendation algorithms for long-term rewards

While recommender systems often have recurring interactions with individual users across months or years, the machine learning systems that power personalized recommendations are often optimized for short-term measures of success. For instance, it is common in the industry to train machine learning algorithms that predict the chance a user will (briefly) engage with content if it is shown and then to display whichever content has the greatest predicted chance of engagement.

Maystre et al. (2023) detail efforts to optimize key components of Spotify's recommender systems for personal listening journeys that unfold over months. Beginning with a fairly general reinforcement learning (RL) model, they derive a pragmatic solution by imposing structural assumptions on certain value functions (or Q-functions). They describe the motivation of their modeling as follows:

> *Our approach to modeling the Q-function is driven by a key hypothesis recommendations significantly contribute to long-term user satisfaction by fostering the formation of specific, recurring engagement patterns with individual pieces of content, which we call "item-level listening habits". In the context of audio streaming, these habits can manifest in various forms: a user might develop a regular routine with a specific podcast show, returning to it as new episodes are released; form an attachment to a particular creator's content; or integrate a curated playlist into their daily activities. Here, we use the term "item" to refer to any distinct piece of content that can be recommended, such as a podcast show, an album, or a playlist.*

Their approach ultimately breaks down a complex RL problem into the problem of building two models of the affinity between a user and item:

1. A pre-existing short-term model predicts the chance a user will briefly engage with an item if it is recommended to them. We refer to this as a *clickiness* score.

2. A long-term *stickiness* model forecasts the strength of users' future listening habits with an item as a function of features of the item, features of the user, and the user's

historical engagement with that specific item. In particular, the final implementation predicts 60-day return days: the number of days over the next 60 days in which a user will return to listen to the item.

The researchers derive an overall score for each potential recommendation by combining the predictions of clickiness and stickiness models. Repeated A/B tests show this methodology helps foster lasting relationships between users and creators and leads to increased engagement with the app. At the time of their publication, this approach was being used in production in several parts of the app.

Our work addresses a major shortcoming of the stickiness models: they exacerbate the challenge of rapidly evaluating fresh content. Directly observing whether users tend to "stick to" a newly released podcast show requires waiting 60 days after they first discover the show. This introduces a delay of at least 60 days before the models can identify that newly released content is unusually sticky (and for whom it is sticky). We leverage progressive feedback to accelerate inferences about the stickiness of newly released content.

8.2 A specific recommendation problem: ranking a shelf of discovery recommendations

The methodology we describe now governs podcast recommendations in several parts of the app, but we focus here on results from an A/B test that intervenes on the ranking logic for a specific home page shelf. Figure 10 shows an example of such a recommendation shelf, which helps users discover new content by displaying only podcast shows they haven't listened to before. Like many other personalized recommendation systems, content selection for this shelf follows a two-step process: first, a candidate generation step filters millions of podcast shows down to a manageable personalized list; then, a ranking step determines the order in which these candidates are displayed. The ranking step has a large impact on the likelihood that a recommendation is seen (or "impressed") by the user. Our A/B test intervenes only on this ranking logic, leaving the candidate generation step unchanged.

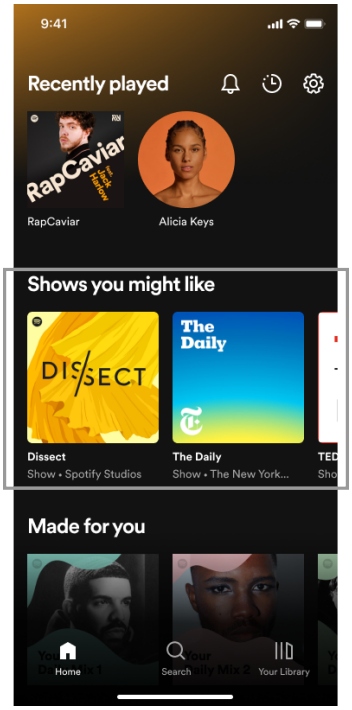

Figure 10: Screenshot of Spotify mobile application. The highlighted podcast discovery shelf contains up to 20 cards that the user can scroll through.

8.3 Some metrics of interest

We now define the metrics used in our algorithms and results. It is worth emphasizing that the first four metrics, around discoveries and their durability, relate to user behavior that could happen anywhere on the app. For instance, a user can discover an item by searching actively for it.

Only the final metric attempts to directly link user behavior to actions of the recommender system.

Discovery When a user listens to part of an episode of a previously undiscovered podcast show, they are said to have "discovered" the podcast. On the "Shows you might like" shelf, all recommended podcasts are new to the user.

60-day minutes (following a discovery) The total number of minutes spent listening to the podcast during the 60 days following a discovery (includes the day it was discovered).

60-day return days (following a discovery) The number of days a user returns to the podcast during the 60-day period after discovering it, excluding the discovery day itself; this metric takes values in $[0, 59]$.

60-day active days (following a discovery) The total number of days a user listens to the podcast during the 60-day period following the discovery (includes the day it was discovered). Equal to return days plus one.

Impression A user has an "impression" of a podcast recommendation when the recommendation appears on their screen. A user viewing the screen in Figure 10 has an impression of two shows on the "Shows you might like" shelf, with additional impressions occurring as they scroll right on the shelf.

60-day engagement attributable to an impression When an impression leads directly to a discovery (through clicking and listening to that recommendation), the subsequent 60-day minutes are *attributable* to the impression. If an impression does not directly result in a discovery, zero minutes are attributable, even if the user later discovers the item through another source. The 60-day return days and 60-day active days attributable to an impression are defined analogously.

In evaluating recommendation efficacy across users, we primarily measure success on a *per-impression* basis, such as discoveries per impression or 60-day minutes per impression. We also examine metrics that focus on the durability of discoveries, such as 60-day return days *per-discovery*.

8.4 Recommendation Algorithm Variants

In the A/B test, each user is randomized to one of the following recommendation ranking algorithms for the duration of the test. Note that the algorithm variants all employ deterministic ranking strategies. Each algorithm is periodically refit using available batch data on the same regular cadence (e.g. daily).

(1) Legacy Approach: Optimizing for the Number of Discoveries. The legacy approach aimed to maximize the probability that users would "discover" a new podcast given an impression. A machine learning model was trained to fit these probabilities from features of the user and item, including powerful learned embeddings of each. Items in the shelf were ranked in descending order of these probabilities. This approach can be viewed as an effort to maximize a short-term proxy reward. Note that we do not directly compare to

these legacy approaches in this work; previous work has already demonstrated the benefit of modeling a long-term reward, as opposed to using discoveries as a proxy reward (Maystre et al., 2023). The approach to modeling long-term rewards taken in this previous work serves as the control policy we compare to in the A/B test (discussed next).

(2) Control Policy: Delayed Rewards Model of Stickiness. The control policy used in our A/B test was learned using the approach introduced in Maystre et al. (2023). For discovery recommendations, the authors suggest scoring a potential recommendation to a user via the metric

$$\texttt{score} = \underbrace{P(\text{discovery} \mid \text{impression})}_{\text{clickiness}} \times \underbrace{\mathbb{E}\left[\text{60-day active days} \mid \text{discovery}\right]}_{\text{stickiness}}, \tag{14}$$

which equals the expected number of 60-day active days attributable to an impression. The personalized probability of a discovery given an impression was modeled using the legacy approach. Through the logic above, this is augmented by incorporating personalized stickiness estimates. For every given item a, they gather a dataset

$$\mathcal{D}_a = \left\{ \left(X_u,\, Y_u := (Y_u^{(1)}, \ldots, Y_u^{(60)}),\, R_u := \sum_{j=1}^{60} Y_u^{(j)} \right) \right\},$$

which ranges over users u who discovered item a at least 60 days ago. Discoveries that happened anywhere on the app over a long time-span are pooled together. For each such user, the dataset contains a tuple consisting of a pre-trained vector representation of user tastes X_u, an activity trace Y_u where $Y_u^{(j)} \in \{0, 1\}$ indicates whether user u listened to the item j days post-discovery, and the *post-discovery* engagement $R_u = \sum_{j=1}^{60} Y_u^{(j)}$ equal to the total number of active days within 60 days. Stickiness is estimated through a linear model

$$\text{stickiness}(a, u) = \theta_a^\top X_u \quad \text{where} \quad \theta_a = \text{argmin}_\theta \sum_{(X,Y,R)\in\mathcal{D}_a} (X^\top\theta - R)^2. \tag{15}$$

We refer to a θ_a as the stickiness vector (or stickiness embedding) of item a. Using observed data on post-discovery listening habits enables learning of items that are unusually "sticky" (and for whom they are sticky). Note, however, that this estimation procedure cannot learn from discoveries that occurred within the last 60 days. Newer content cannot be evaluated, and the system defaults to a fixed, content-agnostic stickiness in this case. This is the main limitation we seek to address.

(3) Treatment Policy: Progressive Feedback Model of Stickiness. The treatment policy in the A/B test uses progressive feedback to draw faster inferences about items' stickiness vectors θ_a; all other aspects of the algorithm, including the scoring formula (14) and clickiness model are the same as in the control policy. Observe that the post-discovery engagement $R_u = \sum_{j=1}^{60} Y_u^{(j)}$ is only fully observable after a 60-day delay, but the activity trace $Y_u = (Y_u^{(1)}, \ldots, Y_u^{(60)})$ on which it depends is revealed progressively. We apply a contextual generalization of the Gaussian filtering algorithm (see Appendix C) to update beliefs about an item's stickiness vectors θ_a from censored activity traces. Potential recommendations are scored using the posterior mean of stickiness predictions in the score (14), and, like

Table 1: In the Spotify A/B test, we observe no substantial difference in the volume of recommendations, recent or otherwise, across cells.

	All shows	Recently-released shows only
Relative difference, treatment vs. control	−0.02%	+2.15%

the control policy, items are ranked in decreasing order of their (personalized) scores. The A/B test isolates the benefit of incorporating progressive feedback into stickiness estimates, showing large benefits even in a highly complex industrial-scale system.

8.5 A/B Test Results

The use of progressive feedback is especially important when recommending recently released shows, because they lack of historical data and since there are fewer long-term engagement outcomes observed due to the delay structure. For this reason, we discuss the A/B test results at two scales: impact when recommending recently released shows versus impact on the quality of recommendations across all podcasts.

8.5.1 Efficacy When Recommending Recently Released Shows

Table 1 and Figure 11 (right) summarize the effects of the A/B test. The key findings are as follows:

1. **No change in the volume of recommendations of recent releases.** The total number of impressions of recently released shows is virtually the same between control and treatment groups. The treatment policy alters *which* recent shows are recommended and *to whom* they are recommended, while keeping the overall frequency of recent show recommendations unchanged.

2. **Huge increases in the quality of recommendations of recent releases.** When recommending recently released shows, the discovery rate (or number of discoveries per impression) was nearly 30% higher under treatment than control. These discoveries were also more lasting, with 60-day active days per-impression, 60-day minutes per-impression, and 60-day return days per-impression increased by over 50% as compared to control.[2] Note that these are *huge* improvements in the context of A/B testing (Zhang et al., 2023; Azevedo et al., 2020, 2019; Rosenthal et al., 1994).

Evidence that incorporating progressive feedback improved long-term recommendations is further supported by Figure 12, where we examine how the recommendation policy increased the 60-day activity per recent show impression. In hindsight[3], we have sufficient data to accurately estimate the average (unpersonalized) long-term value of each show that was

2. The metric 60-day minutes per-impression is equal to the total number of 60-day minutes attributable to impressions during the test (summed over all impressions of recent releases on the shelf by any user), divided by the number of impressions.
3. The test results are being written up 6 months after the test.

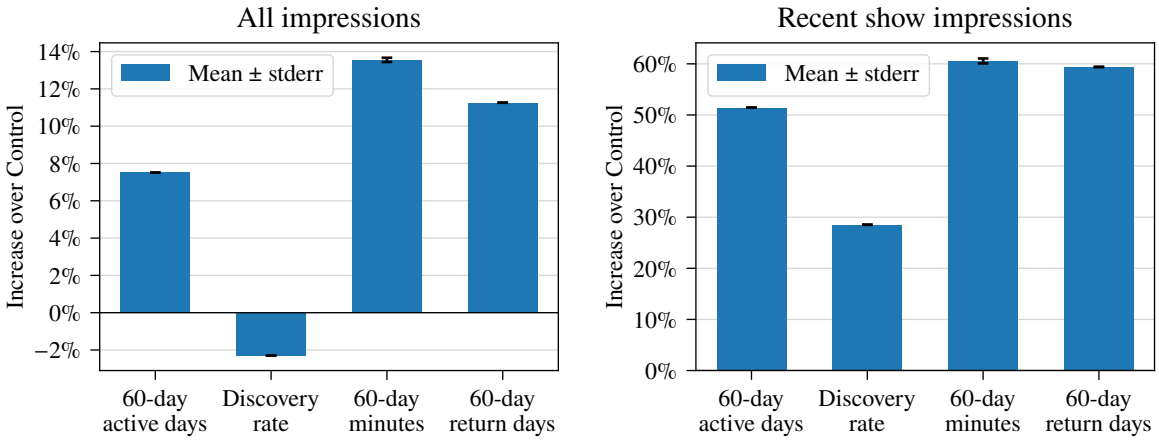

Figure 11: Results of treatment policy compared to control policy for recommending all shows (left) and for recently-released shows only (right). Note that all metrics are computed using only engagement attributable to an impression (as defined in Section 8.3).

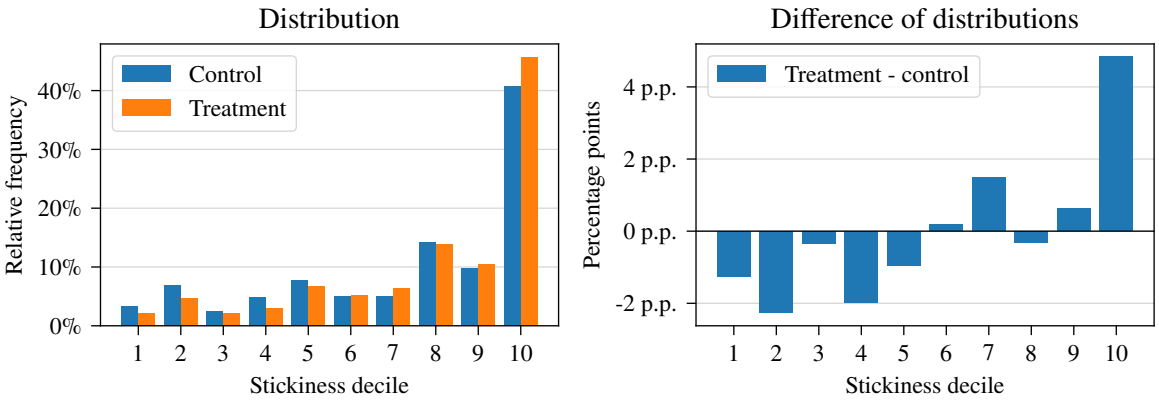

Figure 12: Stickiness of discoveries of recently released shows.

recent during the test period. We leverage this data in Figure 12, partitioning recent shows by their average long-term stickiness (measured in hindsight).

The analysis reveals that, relative to the control policy (delayed rewards), the treatment policy (progressive feedback) decreased impressions to less sticky shows, particularly among the 50% least sticky shows. Instead, the algorithm allocated these impressions to highly sticky shows, increasing the relative fraction of impressions these shows receive. A plausible interpretation is that the treatment policy successfully identified and redistributed impressions based on early indicators of shows' long-term value, which proved to correlate with the shows' eventual stickiness.

8.5.2 Recommendations Beyond Recently Released Shows

We have seen that incorporating progressive feedback into stickiness estimates led to substantial improvements in the quality of recommendations for recently released shows, a critical part of the system's overall promise for users and creators.

The changes also led to meaningful improvements in metrics for the shelf overall, beyond recently released shows. Table 1 and Figure 11 (left) summarize the findings. The number of impressions on the shelf remained virtually identical between treatment and control groups, suggesting that differences in other metrics are primarily attributable to the quality of the personalized recommendations.

Compared to the control algorithm, the treatment algorithm led to a 7% increase in 60-day active days per impression and over 10% increases in both 60-day minutes per impression and 60-day return days per impression. In line with findings in Maystre et al. (2023), these improvements in long-term metrics came at the expense of a slight degradation in short-term metrics: the treatment group had slightly fewer discoveries per impression than the control group.

8.6 Subsequent testing and rollout

Following the initial A/B test, the method underwent further testing in more diverse parts of the app, building trust in the initial findings. First, a subsequent A/B test applied the model to several additional shelves and to recommendations on the podcast sub-feed, allowing for broader impact. Then, a method leveraging our Bayesian filtering procedure was rolled out to all users. The rollout process itself served as another controlled experiment by gradually ramping up the proportion of users receiving the treatment algorithm (Kohavi et al., 2020, Chapter 15).

9. Discussion

In this work, we develop a Thompson sampling based algorithm that can take advantage of intermediate progressive feedback in a setting in which observing the true reward is delayed. We found in both simulation settings and a real A/B test for Spotify podcast recommendations, that the algorithm outperformed methods that do not take advantage of progressive feedback. Given this work, there are several directions for future research, which we discuss below.

The Impatient Bandit algorithm assumes that the prior distribution is learned correctly. A limitation of this work is that we did not provide any guarantees regarding under what settings one would expect to learn an accurate prior from previously collected data. We found in our experiments that we were able to learn an effective prior from data. Recent works have begun to characterize the regret of Thompson Sampling with learned priors (Simchowitz et al., 2021; Zhang et al., 2024).

We discuss how the Impatient Bandit algorithm is able to incorporate user context features, when expected outcomes are linear functions of these contexts (see Appendix C). A limitation of our current algorithm is that we are restricted to linear and kernel type outcome models, which enables us to use Gaussian Bayesian inference methods. An open question is how to computationally efficiently update the posterior distribution for other types of

outcomes. Another direction for future research is to extend our regret analysis to settings with more general outcome models. Finally, it would be of interest to extend our algorithm to handle Markov Decision Process environments with progressive feedback and/or delayed rewards.

Acknowledgements

We thank the reviewers for their thoughtful comments, which helped us improve the paper. We also thank Mounia Lalmas-Roelleke for feedback and discussions on this work.

Overview of Appendices

- Appendix A: Proof of Lemma 2 (Properties of Multivariate Gaussian Model)
- Appendix B: Deriving Incremental Posterior Updates
- Appendix C: Impatient Bandit Algorithm with Context
- Appendix D: Proof of Theorem 1 and Corollary 4 (Regret Bounds)
- Appendix E: Rounding Procedure

Appendix A. Multivariate Gaussian Bayesian Model: Proof of Lemma 2

Lemma 2 (Multivariate Gaussian Bayesian Model)**.** *Let Assumptions 1 and 3 hold. The limit* $\theta_a \triangleq \lim_{|U|\to\infty} \frac{1}{|U|}\sum_{u\in U} Y_u(a)$ *exists almost surely. Conditional on* θ_a, Z_a*, the random variables* $\{Y_u(a)\}_{u\in\mathcal{U}}$ *are i.i.d. Moreover,*

$$\theta_a \mid (Z_a = z) \sim N\left(\mu_{1,z}\,,\, \Sigma_{1,z}\right) \qquad \textit{and} \qquad Y_u(a) \mid (\theta_a, Z_a = z) \sim N(\theta_a\,,\, V_z). \tag{16}$$

Proof As discussed earlier below Remark 1, by exchangeability Assumption 1 and De Finetti's theorem (Heath and Sudderth, 1976; De Finetti, 1937), for any $a \in \mathcal{A}$, there exists some (potentially) random outcome distribution which we denote using P_a that may depend on Z_a that can be used to describe the data generating process for $\{Y_u(a)\}_{u\in U}$:

$$\text{sample } P_a \mid Z_a, \text{ then draw } Y_1(a), Y_2(a), Y_3(a), \ldots \overset{\text{i.i.d.}}{\sim} P_a. \tag{17}$$

Note that $\mathbf{E}[|Y_u(a)| \mid P_a] < \infty$ w.p. 1. If this were not the case, then $\mathbf{E}[|Y_u(a)| \mid P_a]$ could be infinite with non-zero probability, which would imply $\mathbf{E}[|Y_u(a)|] = \mathbf{E}[\mathbf{E}[|Y_u(a)| \mid P_a]]$ is infinite and violate Assumption 3. Thus, we can apply the strong Law of Large Numbers after conditioning on the draw of P_a to get the following result:

$$\mathbb{P}\left(\lim_{|U|\to\infty} \bar{Y}_U(a) = \mathbf{E}\big[Y_u(a) \mid P_a\big] \;\middle|\; P_a, Z_a\right) = 1 \;\text{ w.p. 1},$$

where $\bar{Y}_U(a) \triangleq \frac{1}{|U|}\sum_{u\in U} Y_u(a)$. Since the above holds with probability 1, it implies that

$$\mathbb{P}\left(\lim_{|U|\to\infty} \bar{Y}_U(a) = \mathbf{E}\big[Y_u(a) \mid P_a\big] \;\middle|\; Z_a\right) = 1 \;\text{ w.p. 1}. \tag{18}$$

Thus we have shown that, with probability 1, $\theta_a \triangleq \lim_{|U|\to\infty} \bar{Y}_U(a)$ exists and is equal to $\mathbf{E}\big[Y_u(a) \mid P_a\big]$.

Almost sure convergence implies convergence in distribution. Therefore, $\lim_{|U|\to\infty} \mathcal{L}(\bar{Y}_U(a) \mid Z_a = z) = \mathcal{L}(\theta_a \mid Z_a = z)$ where $\mathcal{L}(X)$ denotes the law (or probability distribution) of a random variable X.

To show (16), we calculate $\mathcal{L}(\bar{Y}_U(a) \mid Z_a = z)$ and then take the limit as $|U| \to \infty$. Note that by Assumptions 1 and 3, the joint distribution of $(Y_u(a))_{u\in U}$ for any U must be of the

following form:

$$\begin{pmatrix} Y_1(a) \\ Y_2(a) \\ Y_3(a) \\ \vdots \\ Y_{|U|}(a) \end{pmatrix} \Bigg| Z_a = z \sim N\left(\begin{bmatrix} \mu_{1,z} \\ \mu_{1,z} \\ \mu_{1,z} \\ \vdots \\ \mu_{1,z} \end{bmatrix}, \begin{bmatrix} \Sigma_{1,z} + V_z & \Sigma_{1,z} & \Sigma_{1,z} & \dots & \Sigma_{1,z} \\ \Sigma_{1,z} & \Sigma_{1,z} + V_z & \Sigma_{1,z} & \dots & \Sigma_{1,z} \\ \Sigma_{1,z} & \Sigma_{1,z} & \Sigma_{1,z} + V_z & \dots & \Sigma_{1,z} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Sigma_{1,z} & \Sigma_{1,z} & \Sigma_{1,z} & \dots & \Sigma_{1,z} + V_z \end{bmatrix} \right)$$

Using that the sum of jointly Gaussian random variables is Gaussian and linearity properties of Gaussian random variables, we find,

$$\bar{Y}_U \mid Z_a = z \sim N\left(\mu_{1,z}, \frac{(|U|-1)}{|U|}\Sigma_{1,z} + \frac{1}{|U|}V_z\right).$$

Thus, $\lim_{|U|\to\infty} \mathcal{L}\left(\bar{Y}_U \mid Z_a = z\right) = N(\mu_{1,z}, \Sigma_{1,z})$. Thus, we have shown that $\theta_a \sim N(\mu_{1,z}, \Sigma_{1,z})$.

One can generalize the above argument to show that $\begin{pmatrix} \theta_a \\ \{Y_u(a)\}_{u\in\mathcal{U}} \end{pmatrix} \Big| Z_a = z$ is a Gaussian process. Specifically, for any $u, u' \in \mathcal{U}$ with $u \neq u'$,

$$\begin{pmatrix} \theta_a \\ Y_u(a) \\ Y_{u'}(a) \end{pmatrix} \Bigg| Z_a = z \sim N\left(\begin{bmatrix} \mu_{1,z} \\ \mu_{1,z} \\ \mu_{1,z} \end{bmatrix}, \begin{bmatrix} \Sigma_{1,z} & \Sigma_{1,z} & \Sigma_{1,z} \\ \Sigma_{1,z} & \Sigma_{1,z} + V_z & \Sigma_{1,z} \\ \Sigma_{1,z} & \Sigma_{1,z} & \Sigma_{1,z} + V_z \end{bmatrix} \right)$$

By properties of conditioning on Gaussian random variables,

$$\begin{pmatrix} Y_u(a) \\ Y_{u'}(a) \end{pmatrix} \Bigg| (Z_a = z, \theta_a) \sim N\left(\begin{bmatrix} \theta_a \\ \theta_a \end{bmatrix}, \begin{bmatrix} V_z & 0 \\ 0 & V_z \end{bmatrix} \right)$$

By the above, we have that $Y_u(a) \mid (\theta_a, Z_a = z) \sim N(\theta_a\,, V_z)$. Moreover, $Y_u(a) \perp\!\!\!\perp Y_{u'}(a) \mid (Z_a = z, \theta_a)$, since for jointly Gaussian random variables, zero correlation implies independence. Moreover, under Assumption 1, $\{Y_u(a)\}_{u\in\mathcal{U}}$ are exchangeable conditional on Z_a, thus we have that conditional on θ_a, Z_a, the random variables $\{Y_u(a)\}_{u\in U}$ are i.i.d. ■

Appendix B. Incremental Posterior Updates

Since the outcomes $Y_u^{(1)}, Y_u^{(2)}, \dots, Y_u^{(J)}$ are revealed incrementally over time after selecting action $A_u = a$, the posterior distribution for θ_a must accordingly be updated with this incremental information. In this Gaussian case, there is a closed form formula for the posterior distribution. Moreover, the posterior updates can happen incrementally, as we get more information on the outcomes $Y_u^{(1)}, Y_u^{(2)}, \dots, Y_u^{(J)}$.

Updating the posterior distribution in this setting is slightly different from standard Gaussian bandit setting because the outcomes $Y_u^{(1)}, Y_u^{(2)}, \dots, Y_u^{(J)}$ from a single user u over time are correlated according to the covariance matrix V_z (see the end of Section 3.2). Specifically, the dimensions of the "noise" vector $Y_u - \theta_a \in \mathbb{R}^J$ are correlated according to V_z:

$$Y_u - \theta_a \mid (A_u = a, Z_a = z) \sim N(0, V_z) \quad \text{where} \quad \theta_a \mid (Z_a = z) \sim N(\mu_{1,z}, \Sigma_{1,z}).$$

Incremental updating of the posterior can be done computationally inexpensively with a clever use of the Cholesky Decomposition (Haddad, 2009). Assume the covariance matrix V_z is invertible. Otherwise, one of the J engagement observations provides redundant information and could be ignored without loss. By the Cholesky decomposition, since V_z is a symmetric positive definite matrix, $V_z = L_z L_z^\top$ for a unique lower triangular matrix $L_z \in \mathbb{R}^{J\times J}$. The inverse matrix L_z^{-1} is also lower triangular. Let $\ell_z^{(j)}$ denotes row j of L_z^{-1}. Then define the transformed engagement observations,

$$\mathcal{Y}_u = \begin{pmatrix} \mathcal{Y}_u^{(1)} \\ \mathcal{Y}_u^{(2)} \\ \vdots \\ \mathcal{Y}_u^{(J)} \end{pmatrix} \triangleq \begin{pmatrix} (\ell_z^{(1)})^\top Y_u \\ (\ell_z^{(2)})^\top Y_u \\ \vdots \\ (\ell_z^{(J)})^\top Y_u \end{pmatrix} = L_z^{-1} Y_u. \tag{19}$$

Working with the transformed vectors provides a convenient expression for the posterior (see Proposition 6 below). The formula is analogous to ones that appear in treatments of Bayesian linear regression (Agrawal and Goyal, 2013; Rasmussen et al., 2006), where $\ell_z^{(1)}, \ldots, \ell_z^{(J)}$ act as fixed "context feature vectors" derived from the prior covariance V_z.

A subtlety in deriving this result is that entries of Y_u are observed progressively, rather than all at once. The proof shows that, since both L_z and L_z^{-1} are lower triangular, the first j entries of $\mathcal{Y}_u$ can be computed from the first j entries of Y_u, and vice versa. The components of the transformed engagements $\mathcal{Y}_u$ become 'uncensored' at the same time as the raw engagements Y_u and provide identical information to the system. A detailed proof is given in Appendix B.

Algorithm 3 Posterior Update

Inputs: $\left\{\mu_{t,a}, \Sigma_{t,a}, V_{Z_a}\right\}_{a\in\mathcal{A}}$, $\left\{\tau_u, A_u, \tilde{Y}_{u,t}\right\}_{u\in\mathcal{U}_{\le t}}$ ▷ Posterior parameters, Previous data

for $a \in \mathcal{A}$ **do**

For all $u \in \mathcal{U}_{\le t}$ and $j \in [1\colon J]$ such that $t = \tau_u + d_j$, compute

$$\mathcal{Y}_u^{(j)} \leftarrow (\ell_z^{(j)})^\top \begin{bmatrix} Y_u^{(1:j)} \\ \mathbf{0}_{J-j} \end{bmatrix} \quad \text{where } \mathbf{0}_{J-j} \in \mathbb{R}^{J-j} \text{ is a vector of zeros}$$

$\Sigma_{t+1,a} \leftarrow \left(\Sigma_{t,a}^{-1} + \sum_{u\in\mathcal{U}_{\le t}} \mathbb{1}\{A_u = a\} \sum_{j=1}^{J} \mathbb{1}\{t = \tau_u + d_j\} \ell_z^{(j)} (\ell_z^{(j)})^\top\right)^{-1}$

$\mu_{t+1,a} \leftarrow \Sigma_{t+1,a}\left(\Sigma_{t,a}^{-1}\mu_{t,a} + \sum_{u\in\mathcal{U}_{\le t}} \mathbb{1}\{A_u = a\} \sum_{j=1}^{J} \mathbb{1}\{t = \tau_u + d_j\} \ell_z^{(j)} \mathcal{Y}_u^{(j)}\right)$

Output: $\{\mu_{t+1,a}, \Sigma_{t+1,a}\}_{a\in\mathcal{A}}$ ▷ Posterior parameters

Proposition 6 (Posterior Distribution)**.** *Let Assumptions 1 and 3 hold, and let V_z and $\Sigma_{0,z}$ be positive definite. Then, $\theta_a \mid \mathcal{H}_t \sim N(\mu_{t,a}, \Sigma_{t,a})$ where*

$$\Sigma_{t,a} \triangleq \left(\Sigma_{1,z}^{-1} + \sum_{u\in\mathcal{U}_{<t}} \mathbb{1}\{A_u = a\} \sum_{j=1}^{J} \mathbb{1}\{t > \tau_u + d_j\} \ell_z^{(j)} (\ell_z^{(j)})^\top\right)^{-1}$$

and

$$\mu_{t,a} \triangleq \Sigma_{t,a}\Big(\Sigma_{1,z}^{-1}\mu_{1,z} + \sum_{u\in\mathcal{U}_{<t}} \mathbb{1}\{A_u = a\}\sum_{j=1}^{J} \mathbb{1}\{t > \tau_u + d_j\}\ell_z^{(j)}\mathcal{Y}_u^{(j)}\Big).$$

Proof It is sufficient to show that if $\theta_a \mid (Z_a = z) \sim N(\mu_{1,z}, \Sigma_{1,z})$ then, for any $k \in [1\colon J]$,

$$\theta_a \mid \big(Y_u^{(1)}, Y_u^{(2)}, \ldots, Y_u^{(k)}, A_u = a\big) \sim N\left(\mu_{\text{post}}, \Sigma_{\text{post}}\right)$$

where for $\mathcal{Y}_u^{(j)}$ defined as in (19),

$$\Sigma_{\text{post}} \triangleq \left\{\Sigma_{1,z}^{-1} + \sum_{j=1}^{k} \ell_z^{(j)}(\ell_z^{(j)})^\top\right\}^{-1} \quad \text{and} \quad \mu_{\text{post}} \triangleq \Sigma_{\text{post}}\Big(\Sigma_{1,z}^{-1}\mu_{1,z} + \sum_{j=1}^{k} \ell_z^{(j)}\mathcal{Y}_u^{(j)}\Big). \tag{20}$$

Since the covariance matrix V_z is invertible, by the Cholesky decomposition there is a unique, positive definite lower triangular matrix $L_z \in \mathbb{R}^{J\times J}$ such that $V_z = L_z L_z^\top$. Using the matrix L_z, we can define a version of Y_u, whose noise covariance matrix is "de-correlated":

$$\mathcal{Y}_u \triangleq L_z^{-1}Y_u = L_z^{-1}\big(Y_u - \theta_a\big) + L_z^{-1}\theta_a.$$

Above $\theta_a \mid (Z_a = z) \sim N(\mu_{1,z}, \Sigma_0)$ and $L_z^{-1}\big(Y_u - \theta_a\big) \mid (A_u = a, Z_a = z, \theta_a) \sim N\big(0, I_J\big)$ where $I_J \in \mathbb{R}^{J\times J}$ is the identity matrix; note that $L_z^{-1}V_z(L_z^{-1})^\top = I_J$ since $V_z = L_z L_z^\top$ by definition.

Thus $\mathcal{Y}_u^{(j)}$, the j^{th} element of the vector $\mathcal{Y}_u$, can be written as

$$\mathcal{Y}_u^{(j)} \triangleq (\ell_z^{(j)})^\top Y_u \underbrace{=}_{(a)} (\ell_z^{(j)})^\top \begin{bmatrix} Y_u^{(1:j)} \\ \mathbf{0}_{J-j} \end{bmatrix} \quad \text{where} \quad \mathcal{Y}_u^{(j)} \mid (A_u = a, Z_a = z, \theta_a) \sim N\big((\ell_z^{(j)})^\top\theta_a,\ 1\big).$$

Above, $(\ell_z^{(j)})^\top$ is the j^{th} row of the matrix L_z^{-1}. We now explain why equality (a) above holds. By the lower-triangular structure of the non-singular matrix L_z (due to the Cholesky composition), the inverse matrix L_z^{-1} is also lower-triangular (Meyer, 1970). Thus, the last $J - j$ dimensions of $(\ell_z^{(j)})^\top$, the j^{th} row of L_z^{-1}, must all be zeros.

Since $\theta_a \mid (Z_a = z) \sim N(\mu_{1,z}, \Sigma_{1,z})$ and $\mathcal{Y}_u^{(j)} \mid (A_u = a, Z_a = z, \theta_a) \sim N\big((\ell_z^{(j)})^\top\theta_a,\ 1\big)$ by the display above, by conditioning properties of multivariate Gaussians (see Section 2.2 and Appendix A.1 in Agrawal and Goyal (2013) for details), we have

$$\theta_a \mid \big(\mathcal{Y}_u^{(1)}, \mathcal{Y}_u^{(2)}, \ldots, \mathcal{Y}_u^{(k)}, A_u = a\big) \sim N\left(\mu_{\text{post}}, \Sigma_{\text{post}}\right),$$

for μ_{post} and Σ_{post} as defined in (20).

The final step is to show that the following are equal in distribution:

$$\theta_a \mid \big(\mathcal{Y}_u^{(1)}, \mathcal{Y}_u^{(2)}, \ldots, \mathcal{Y}_u^{(k)}, A_u = a, Z_a = z\big) \quad \stackrel{D}{=} \quad \theta_a \mid \big(Y_u^{(1)}, Y_u^{(2)}, \ldots, Y_u^{(k)}, A_u = a, Z_a = z\big).$$

Note it is sufficient to show that $\mathcal{Y}_u^{(1:k)} \triangleq \big(\mathcal{Y}_u^{(1)}, \mathcal{Y}_u^{(2)}, \ldots, \mathcal{Y}_u^{(k)}\big)$ is a one-to-one (injective), non-random transformation of $Y_u^{(1:k)} \triangleq \big(Y_u^{(1)}, Y_u^{(2)}, \ldots, Y_u^{(k)}\big)$. We prove this below.

Since $\mathcal{Y}_u^{(j)} = (\ell_z^{(j)})^\top \begin{bmatrix} Y_u^{(1:j)} \\ \mathbf{0}_{J-j} \end{bmatrix}$, we have that $\begin{bmatrix} \mathcal{Y}_u^{(1:k)} \\ \mathbf{0}_{J-k} \end{bmatrix} = L_z^{-1} \begin{bmatrix} Y_u^{(1:k)} \\ \mathbf{0}_{J-k} \end{bmatrix}$. It is sufficient to show that the function $f(x) = L_z^{-1}x$ for any vector $x \in \mathbb{R}^J$ is injective. Suppose that $f(x) = f(x')$, i.e., $L_z^{-1}x = L_z^{-1}x'$. Since L_z^{-1} is positive definite, by multiplying both sides by L_z it must be that $x = x'$. This f is an injective function.

■

Appendix C. Impatient Bandit Algorithm with Context

In this section, we write the impatient bandit algorithm extension to linear contextual bandits. We introduce new notation and redefine previous notation that will be used in this section alone. Each user $u \in \mathcal{U}$ has an associated context vector $X_u \in \mathbb{R}^k$ that is observed prior to making the recommendation decision. The algorithm posits the following data generating process for each item $a \in \mathcal{A}$:

$$\theta_a \mid (Z_a = z) \sim N\left(\mu_{1,z}\,,\, \Sigma_{1,z}\right)$$

Above $\theta_a, \mu_{1,z} \in \mathbb{R}^{k \cdot J}$ and $\Sigma_{1,z} \in \mathbb{R}^{(k \cdot J) \times (k \cdot J)}$. We use the notation $\theta_a = \left(\theta_a^{(1)}, \theta_a^{(1)}, \ldots, \theta_a^{(J)}\right)$ where each $\theta_a^{(j)} \in \mathbb{R}^k$. Similarly, we will use the notation $\mu_{1,z} = \left(\mu_{1,z}^{(1)}, \mu_{1,z}^{(2)}, \ldots, \mu_{1,z}^{(J)}\right)$ where each $\mu_{1,z}^{(j)} \in \mathbb{R}^k$. Potential outcomes are now distributed as follows:

$$Y_u(a) \mid (\theta_a, X_u, Z_a = z) \sim N\left(\begin{bmatrix} X_u^\top \theta_a^{(1)} \\ \vdots \\ X_u^\top \theta_a^{(J)} \end{bmatrix}\,,\, V_z\right).$$

Above $V_z \in \mathbb{R}^{J \times J}$. We use $Y_u(a) = \left(Y_u^{(1)}(a), Y_u^{(2)}(a), \ldots, Y_u^{(J)}(a)\right) \in \mathbb{R}^J$. We continue to use $\tilde{Y}_{u,t}^{(j)}$ to refer to the censored outcomes, as defined in (2). We now redefine the history to include the past observed user contexts X_u:

$$\mathcal{H}_t \triangleq \bigcup_{u \in \mathcal{U}_{<t}} \left\{A_u, X_u, \tilde{Y}_{u,t}^{(1)}(A_u), \tilde{Y}_{u,t}^{(2)}(A_u), \ldots, \tilde{Y}_{u,t}^{(J)}(A_u)\right\} \cup \{Z_a : a \in \mathcal{A}\}.$$

Similar to the non-contextual case, we assume the covariance matrix V_z is invertible. Since the covariance matrix V_z is invertible, by the Cholesky decomposition there is a unique, positive definite lower triangular matrix $L_z \in \mathbb{R}^{J \times J}$ such that $V_z = L_z L_z^\top$. We use $(\ell_z^{(j)})^\top$ to refer to the j^{th} row of the matrix L_z^{-1}. Furthermore, we use $\ell_z^{(j,i)}$ to refer to the i^{th} element of the vector $(\ell_z^{(j)})^\top$. The posterior update formula for the contextual version of impatient bandits will use a new featurization function of both the user context X_u and vector $(\ell_z^{(j)})^\top$:

$$\phi\left(X_u, \ell_z^{(j)}\right) \triangleq \begin{bmatrix} X_u \ell_z^{(j,1)} \\ \vdots \\ X_u \ell_z^{(j,J)} \end{bmatrix} \in \mathbb{R}^{J \cdot k} \tag{21}$$

Algorithm 4 Impatient Bandit Thompson Sampling with Context π^{TS}

Require: Priors and noise covariance matrices $\left\{\mu_{1,Z_a}, \Sigma_{1,Z_a}, V_{Z_a}\right\}_{a\in\mathcal{A}}$

for $a \in \mathcal{A}$ **do**
  $(\mu_{1,a}, \Sigma_{1,a}) \leftarrow (\mu_{1,Z_a}, \Sigma_{1,Z_a})$ ▷ Initialize beliefs.
$\mathcal{H}_1 \leftarrow \{Z_a : a \in \mathcal{A}\}$ ▷ Initialize history.
for $t = 1, 2, \ldots, T$ **do**
  for $u \in \mathcal{U}_t$ **do**
    $\tau_u \leftarrow t$ ▷ Set user decision time.
    for $a \in \mathcal{A}$ **do**
      $\theta'_a \sim N\left(\mu_{t,a}\,,\, \Sigma_{t,a}\right)$ ▷ Sample from belief.
    $A_u \leftarrow \mathrm{argmax}_{a\in\mathcal{A}} \left\{R(\theta'_{a,X_u})\right\}$ for $\theta'_{a,X_u} \leftarrow \begin{bmatrix} X_u^\top(\theta'_a)^{(1)} \\ \vdots \\ X_u^\top(\theta'_a)^{(J)} \end{bmatrix}$ ▷ Take action.
  $\mathcal{H}_{t+1} \leftarrow \mathcal{H}_t \cup \left\{Y_u^{(j)}(A_u) : u \in \mathcal{U}_{\le t}, j \text{ s.t. } t = \tau_u + d_j\right\}$ ▷ Update history.
  for $a \in \mathcal{A}$ **do**
    $(\mu_{t+1,a}, \Sigma_{t+1,a}) \leftarrow \mathrm{PosteriorUpdate}(\mu_{t,Z_a}, \Sigma_{t,Z_a}, V_{Z_a}, \mathcal{H}_{t+1})$ ▷ Update belief.

Algorithm 5 Posterior Update with Context

Inputs: $\left\{\mu_{t,a}, \Sigma_{t,a}, V_{Z_a}\right\}_{a\in\mathcal{A}}$, $\left\{\tau_u, A_u, X_u, \tilde{Y}_{u,t+1}\right\}_{u\in\mathcal{U}_{\le t}}$ ▷ Posterior parameters, Previous data
for $a \in \mathcal{A}$ **do**
  For all $u \in \mathcal{U}_{\le t}$ and $j \in [1\colon J]$ such that $t = \tau_u + d_j$, compute

$$\mathcal{Y}_u^{(j)} \leftarrow (\ell_z^{(j)})^\top \begin{bmatrix} Y_u^{(1:j)} \\ \mathbf{0}_{J-j} \end{bmatrix} \quad \text{where } \mathbf{0}_{J-j} \in \mathbb{R}^{J-j} \text{ is a vector of zeros}$$

  $\Sigma_{t+1,a} \leftarrow \left(\Sigma_{t,a}^{-1} + \sum_{u\in\mathcal{U}_{\le t}} \mathbb{1}\{A_u = a\} \sum_{j=1}^J \mathbb{1}\{t = \tau_u + d_j\} \phi(X_u, \ell_z^j)\phi(X_u, \ell_z^j)^\top\right)^{-1}$
  $\mu_{t+1,a} \leftarrow \Sigma_{t+1,a}\left(\Sigma_{t+1,a}^{-1}\mu_{t,a} + \sum_{u\in\mathcal{U}_{\le t}} \mathbb{1}\{A_u = a\} \sum_{j=1}^J \mathbb{1}\{t = \tau_u + d_j\} \phi(X_u, \ell_z^j)\mathcal{Y}_u^{(j)}\right)$
Output: $\{\mu_{t+1,a}, \Sigma_{t+1,a}\}_{a\in\mathcal{A}}$ ▷ Posterior parameters

Posterior Update Formula Derivation We now provide an informal justification for the posterior update formula, which extends the proof of Proposition 6. Using the matrix L_z (recall L_z is lower triangular and $V_z = L_z L_z^\top$), we can define a version of Y_u, whose noise covariance matrix is "de-correlated":

$$\mathcal{Y}_u \triangleq L_z^{-1} Y_u = L_z^{-1}\left(Y_u - \begin{bmatrix} X_u^\top\theta_a^{(1)} \\ \vdots \\ X_u^\top\theta_a^{(J)} \end{bmatrix}\right) + L_z^{-1}\begin{bmatrix} X_u^\top\theta_a^{(1)} \\ \vdots \\ X_u^\top\theta_a^{(J)} \end{bmatrix}.$$

Above $\theta_a \mid (X_u, Z_a = z) \sim N(\mu_{1,z}, \Sigma_{0,z})$ and

$$L_z^{-1}\left(Y_u - \begin{bmatrix} X_u^\top \theta_a^{(1)} \\ \vdots \\ X_u^\top \theta_a^{(J)} \end{bmatrix}\right) \Big| (X_u, A_u = a, Z_a = z, \theta_a) \sim N\big(0, I_J\big)$$

where $I_J \in \mathbb{R}^{J \times J}$ is the identity matrix; note that $L_z^{-1} V_z (L_z^{-1})^\top = I_J$ since $V_z = L_z L_z^\top$ by definition.

Thus $\mathcal{Y}_u^{(j)}$, the j^{th} element of the vector $\mathcal{Y}_u$, can be written as

$$\mathcal{Y}_u^{(j)} \triangleq (\ell_z^{(j)})^\top Y_u \underbrace{=}_{(a)} (\ell_z^{(j)})^\top \begin{bmatrix} Y_u^{(1:j)} \\ \mathbf{0}_{J-j} \end{bmatrix}$$

where

$$\mathcal{Y}_u^{(j)} \mid (X_u, A_u = a, Z_a = z, \theta_a) \sim N\left((\ell_z^{(j)})^\top \begin{bmatrix} X_u^\top \theta_a^{(1)} \\ \vdots \\ X_u^\top \theta_a^{(j)} \\ \mathbf{0}_{J-j} \end{bmatrix},\ 1 \right)$$

Above $(\ell_z^{(j)})^\top$ is the j^{th} row of the matrix L_z^{-1}. We now explain why equality (a) above holds. By the lower-triangular structure of the non-singular matrix L_z (due to the Cholesky composition), the inverse matrix L_z^{-1} is also lower-triangular (Meyer, 1970). Thus, the last $J - j$ dimensions of $(\ell_z^{(j)})^\top$, the j^{th} row of L_z^{-1}, must all be zeros.

Furthermore, note that by taking the transpose and rearranging terms,

$$(\ell_z^{(j)})^\top \begin{bmatrix} X_u^\top \theta_a^{(1)} \\ \vdots \\ X_u^\top \theta_a^{(j)} \\ \mathbf{0}_{J-j} \end{bmatrix} = \sum_{i=1}^{j} (\theta_a^{(i)})^\top X_u \ell_z^{(j,i)} = (\theta_a)^\top \phi\big(X_u, \ell_z^{(j)}\big),$$

where above we use $\ell_z^{(j,i)}$ to refer to the i^{th} element of the vector $\ell_z^{(j)}$. The final equality uses $\theta_a = \big(\theta_a^{(1)}, \theta_a^{(2)}, \ldots \theta_a^{(J)}\big)$ and the featurization defined in (21).

An argument equivalent to that made at the end of the Proposition 6 proof can be used to show that

$$\theta_a \mid \big(\mathcal{Y}_u^{(1:k)}, X_u, A_u = a, Z_a = z\big) \quad \stackrel{D}{=} \quad \theta_a \mid \big(Y_u^{(1:k)}, X_u, A_u = a, Z_a = z\big).$$

Appendix D. Regret Bounds

D.1 Reduction to estimation error

The next lemma establishes a sense in which Thompson sampling always 'exploits' what it knows about arms' performance, attaining low regret once uncertainty is low. Developing appropriate measures of remaining uncertainty is key to the analysis. We adopt one from

Qin and Russo (2023), which looks at the posterior variance of arms' reward distributions averaged according to the optimal action probabilities: $\sum_{a\in\mathcal{A}} p^*_{t,a}\sigma^2_{t,a}$. Critically, this measure allows an arm to be poorly estimated (large $\sigma^2_{t,a}$) if it has a very low chance of being optimal (low $p^*_{t,a}$).

Lemma 7. *Define* $\sigma^2_{t,a} \triangleq \mathrm{Var}(\bar{R}_a \mid \mathcal{H}_t)$. *Recall from* (1) *that* $\bar{R}_a \triangleq \mathbf{E}\left[R\big(Y_u(a)\big) \mid \theta_a\right]$. *Under Assumptions 1-4,*

$$\mathbf{E}\left[\Delta_t(\pi^{\text{TS-rnd}})\right] \leq \sqrt{2\log(|\mathcal{A}|)\mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\sum_{a\in\mathcal{A}} p^*_{t,a}\sigma^2_{t,a}\right]} + \underbrace{\epsilon_{\text{rnd}}|\mathcal{A}|\max_{a\in\mathcal{A}}\mathbf{E}_{\pi^{\text{TS-rnd}}}\left[R\big(Y_u(A^*)\big) - R\big(Y_u(a)\big)\right]}_{=O(\epsilon_{\text{rnd}})}$$

where $p^*_{t,a} = \mathbb{P}(A^* = a \mid \mathcal{H}_t)$.

Proof Note by the definition of $\Delta_t(\pi)$ from (3), for any policy π,

$$\begin{aligned}
\mathbf{E}\left[\Delta_t(\pi) \mid \mathcal{H}_t\right] &= \mathbf{E}\left[\mathbf{E}_\pi\left[\frac{1}{|\mathcal{U}_t|}\sum_{u\in\mathcal{U}_t}\left\{R\big(Y_u(A^*)\big) - R\big(Y_u(A_u)\big)\right\} \;\middle|\; \{\theta_a\}_{a\in\mathcal{A}}, \mathcal{H}_t\right] \middle| \mathcal{H}_t\right] \\
&\underbrace{=}_{(a)} \mathbf{E}_\pi\left[\frac{1}{|\mathcal{U}_t|}\sum_{u\in\mathcal{U}_t}\left\{R\big(Y_u(A^*)\big) - R\big(Y_u(A_u)\big)\right\} \middle| \mathcal{H}_t\right] \\
&= \frac{1}{|\mathcal{U}_t|}\sum_{u\in\mathcal{U}_t}\sum_{a\in\mathcal{A}} p^{\text{TS-rnd}}_{t,a}\mathbf{E}\left[R\big(Y_u(A^*)\big) - R\big(Y_u(a)\big) \mid \mathcal{H}_t\right] \\
&\underbrace{=}_{(b)} \sum_{a\in\mathcal{A}} p^{\text{TS-rnd}}_{t,a}\mathbf{E}\left[R\big(Y_u(A^*)\big) - R\big(Y_u(a)\big) \mid \mathcal{H}_t\right] \quad \text{for any } u \in \mathcal{U}_t
\end{aligned}$$

Above equality (a) holds by law of iterated expectations by averaging over the draw of latent item features $\{\theta_a\}_{a\in\mathcal{A}}$. Equality (b) holds because of exchangeability Assumption 1.

Below, we consider any $u \in \mathcal{U}_t$. By the result above, we have that

$$\begin{aligned}
&\mathbf{E}\left[\Delta_t(\pi^{\text{TS-rnd}}) \mid \mathcal{H}_t\right] = \sum_{a\in\mathcal{A}} p^{\text{TS-rnd}}_{t,a}\mathbf{E}\left[R\big(Y_u(A^*)\big) - R\big(Y_u(a)\big) \mid \mathcal{H}_t\right] \\
&= \sum_{a\in\mathcal{A}} p^{\text{TS}}_{t,a}\mathbf{E}\left[R\big(Y_u(A^*)\big) - R\big(Y_u(a)\big) \mid \mathcal{H}_t\right] + \sum_{a\in\mathcal{A}}\left(p^{\text{TS-rnd}}_{t,a} - p^{\text{TS}}_{t,a}\right)\mathbf{E}\left[R\big(Y_u(A^*)\big) - R\big(Y_u(a)\big) \mid \mathcal{H}_t\right] \\
&\leq \sum_{a\in\mathcal{A}} p^{\text{TS}}_{t,a}\mathbf{E}\left[R\big(Y_u(A^*)\big) - R\big(Y_u(a)\big) \mid \mathcal{H}_t\right] + \sum_{a\in\mathcal{A}}\left|p^{\text{TS-rnd}}_{t,a} - p^{\text{TS}}_{t,a}\right|\mathbf{E}\left[R\big(Y_u(A^*)\big) - R\big(Y_u(a)\big) \mid \mathcal{H}_t\right]
\end{aligned}$$

Above recall that $p^{\text{TS-rnd}}_{t,a}$ is the number of users allocated to action a in round t according to $\pi^{\text{TS-rnd}}$. $p^{\text{TS}}_{t,a}$ is the probability action a is selected at batch t under Thompson sampling π^{TS}. The final inequality above holds because the regret $\mathbf{E}\left[R\big(Y_u(A^*)\big) - R\big(Y_u(a)\big) \mid \mathcal{H}_t\right]$ is non-negative.

We take an expectation of the above expression to get the following inequality:

$$\mathbf{E}\left[\Delta_t(\pi^{\text{TS-rnd}})\right] \leq \underbrace{\sum_{a\in\mathcal{A}} \mathbf{E}_{\pi^{\text{TS-rnd}}}\left[p_{t,a}^{\text{TS}}\mathbf{E}\left[R\big(Y_u(A^*)\big) - R\big(Y_u(a)\big) \mid \mathcal{H}_t\right]\right]}_{(i)}$$
$$+ \underbrace{\sum_{a\in\mathcal{A}} \mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\left|p_{t,a}^{\text{TS-rnd}} - p_{t,a}^{\text{TS}}\right| \mathbf{E}\left[R\big(Y_u(A^*)\big) - R\big(Y_u(a)\big) \mid \mathcal{H}_t\right]\right]}_{(ii)}$$

We will bound each of the two terms above separately.

Bounding term (i) Below we consider any $u \in \mathcal{U}_t$. First, note the following:

$$\sum_{a\in\mathcal{A}} p_{t,a}^{\text{TS}}\mathbf{E}\left[R\big(Y_u(A^*)\big) - R\big(Y_u(a)\big) \mid \mathcal{H}_t\right] = \mathbf{E}\left[R\big(Y_u(A^*)\big) \mid \mathcal{H}_t\right] - \sum_{a\in\mathcal{A}} p_{t,a}^{\text{TS}}\mathbf{E}\left[R\big(Y_u(a)\big) \mid \mathcal{H}_t\right]$$
$$= \mathbf{E}\left[R\big(Y_u(A^*)\big) \mid \mathcal{H}_t\right] - \sum_{a\in\mathcal{A}} p_{t,a}^{*}\mathbf{E}\left[R\big(Y_u(a)\big) \mid \mathcal{H}_t\right]$$

The last equality above holds because $p_{t,a}^* = \mathbb{P}(A^* = a \mid \mathcal{H}_t) = \mathbb{P}(A_u = a \mid \mathcal{H}_t) = p_{t,a}^{\text{TS}}$. Next, using the linearity of the function R (Assumption 3),

$$= \mathbf{E}\left[R\big(Y_u(A^*)\big) \mid \mathcal{H}_t\right] - \sum_{a\in\mathcal{A}} p_{t,a}^{*} R\big(\mu_{t,a}\big) = \mathbf{E}\left[R\big(Y_u(A^*)\big) \mid \mathcal{H}_t\right] - \mathbf{E}\left[R\big(\mu_{t,A^*}\big) \mid \mathcal{H}_t\right]$$
$$= \mathbf{E}\left[R\big(Y_u(A^*)\big) - R\big(\mu_{t,A^*}\big) \mid \mathcal{H}_t\right]$$
$$= \mathbf{E}\left[R\big(\theta_{A^*}\big) - R\big(\mu_{t,A^*}\big) \mid \mathcal{H}_t\right]$$

The final equality above holds by the linearity of function R and the law of iterated expectations.

By taking the expectation on both sides of the expression above, we get that:

$$\sum_{a\in\mathcal{A}} \mathbf{E}_{\pi^{\text{TS-rnd}}}\left[p_{t,a}^{\text{TS}}\mathbf{E}\left[R\big(Y_u(A^*)\big) - R\big(Y_u(a)\big) \mid \mathcal{H}_t\right]\right] = \mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\mathbf{E}\left[R\big(Y_u(A^*)\big) - R\big(\mu_{t,A^*}\big) \mid \mathcal{H}_t\right]\right]$$
$$\leq \sqrt{\mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\sigma_{t,A^*}^2\right] 2\log(|\mathcal{A}|)} = \sqrt{\mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\sum_{a\in\mathcal{A}} p_{t,a}^{*}\sigma_{t,a}^2\right] 2\log(|\mathcal{A}|)}$$

The last inequality above holds by Proposition 1 of Qin and Russo (2023). See the discussion of Proposition 1 of Qin and Russo (2023) for more commentary.

Bounding term (ii) Consider any $u \in \mathcal{U}_t$.

$$\sum_{a\in\mathcal{A}} \mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\left|p_{t,a}^{\text{TS-rnd}} - p_{t,a}^{\text{TS}}\right| \mathbf{E}\left[R\big(Y_u(A^*)\big) - R\big(Y_u(a)\big) \mid \mathcal{H}_t\right]\right]$$
$$\underbrace{\leq}_{(d)} \epsilon_{\text{rnd}} \sum_{a\in\mathcal{A}} \mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\mathbf{E}\left[R\big(Y_u(A^*)\big) - R\big(Y_u(a)\big) \mid \mathcal{H}_t\right]\right]$$
$$\leq \epsilon_{\text{rnd}}|\mathcal{A}| \max_{a\in\mathcal{A}} \mathbf{E}_{\pi^{\text{TS-rnd}}}\left[R\big(Y_u(A^*)\big) - R\big(Y_u(a)\big)\right]$$

Above inequality (d) holds by rounding Assumption 4. ■

D.2 Helpful Lemmas

By Lemma 7, in order to bound regret, it is enough to bound $\mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\sum_{a\in\mathcal{A}} p^*_{t,a}\sigma^2_{t,a}\right]$. Note,

$$\mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\sum_{a\in\mathcal{A}} p^*_{t,a}\sigma^2_{t,a}\right] = \mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\sum_{a\in\mathcal{A}} \mathbb{1}_{A^*=a}\sigma^2_{t,a}\right] = \mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\sum_{a\in\mathcal{A}} \mathbb{1}_{A^*=a} r_1^\top \operatorname{Cov}\left(\theta_a \mid \mathcal{H}_t\right) r_1\right].$$

Furthermore, note that

$$\begin{aligned}\operatorname{Cov}\left(\theta_a \mid \mathcal{H}_t\right) &= \left\{\Sigma^{-1}_{1,Z_a} + \sum_{u\in\mathcal{U}_{<t}} \mathbb{1}_{A_u=a}\sum_{j=1}^{J}\mathbb{1}\{t > \tau_u + d_j\}\ell^{(j)}_{Z_a}(\ell^{(j)}_{Z_a})^\top\right\}^{-1}\\ &= \left\{\Sigma^{-1}_{1,Z_a} + m\sum_{t'=1}^{t-1} p^{\text{TS-rnd}}_{t',a}\sum_{j=1}^{J}\mathbb{1}\{t > t' + d_j\}\ell^{(j)}_{Z_a}(\ell^{(j)}_{Z_a})^\top\right\}^{-1}\end{aligned}$$

The final equality above holds by re-indexing because $p^{\text{TS-rnd}}_{t',a} = \frac{1}{m}\sum_{u\in\mathcal{U}_t}\mathbb{1}(A_u = a)$ and because if $u \in \mathcal{U}_{t'}$ then $\tau_u = t'$.

The following result (Lemma 8 below) is used to bound $\operatorname{Cov}\left(\theta_a \mid \mathcal{H}_t\right)$. This bound will use $C^{\text{Full}}_{t,a}$, where

$$C^{\text{Full}}_{t,a} \triangleq \operatorname{Cov}\left(\theta_a \mid Z_a, \{\tilde{Y}_{u,t}(a)\}_{u\in\mathcal{U}_{<t}}\right) = \left\{\Sigma^{-1}_{1,Z_a} + m\sum_{t'=1}^{t-1}\sum_{j=1}^{J}\mathbb{1}\{t > t' + d_j\}\ell^{(j)}_{Z_a}(\ell^{(j)}_{Z_a})^\top\right\}^{-1}. \tag{22}$$

We can interpret $C^{\text{Full}}_{t,a}$ as the covariance of θ_a given the data up to time t in a "full feedback" setting, and interpret $\operatorname{Cov}\left(\theta_a \mid \mathcal{H}_t\right)$ as the covariance of θ_a given the data up to time t in a "bandit feedback" setting. Specifically, the difference between $\operatorname{Cov}\left(\theta_a \mid \mathcal{H}_t\right)$ and $C^{\text{Full}}_{t,a}$ is that $\operatorname{Cov}\left(\theta_a \mid \mathcal{H}_t\right)$ only conditions on the outcomes $\left\{\tilde{Y}_{u,t}(a) : A_u = a\right\}_{u\in\mathcal{U}_{<t}}$, according to $\mathcal{H}_t$. In contrast, $C^{\text{Full}}_{t,a}$ conditions on all of $\left\{\tilde{Y}_{u,t}(a)\right\}_{u\in\mathcal{U}_{<t}}$.

Lemma 8 (Lemma 8 of Qin and Russo (2023))**.** *Under Assumptions 1 and 3, for any t and $a \in \mathcal{A}$, with probability* 1*,*

$$\operatorname{Cov}\left(\theta_a \mid \mathcal{H}_t\right) \preceq C^{\text{Full}}_{t,a}\left(\Sigma^{-1}_{1,Z_a} + m\sum_{t'=1}^{t-1}\frac{\sum_{j=1}^{J}\mathbb{1}\{t > t' + d_j\}\ell^{(j)}_{Z_a}(\ell^{(j)}_{Z_a})^\top}{p^{\text{TS-rnd}}_{t',a}}\right)C^{\text{Full}}_{t,a}. \tag{23}$$

The upper bound in Lemma 8 above relates $\operatorname{Cov}\left(\theta_a \mid \mathcal{H}_t\right)$ to an expression that involves $C^{\text{Full}}_{t,a}$ on the right-hand-side of (23). The Lemma follows from an identical proof to the argument in Qin and Russo (2023, Lemma 8).

In general, the inverse propensity weights $1/p^{\text{TS-rnd}}_{t',a}$ in (23) can be extremely large. Thankfully, by Lemma 7 the term we need to bound, $\mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\sigma^2_{t,A^*}\right] = \mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\sum_{a\in\mathcal{A}} p^*_{t,a}\sigma^2_{t,a}\right]$, depends on only the posterior variance *evaluated at the optimal arm* A^*. The next lemma controls the inverse propensity assigns to the optimal arm, $1/p^{\text{TS-rnd}}_{t',A^*}$.

Lemma 9. *Under Assumptions 1, 3, and 4, for any $t \in [1\colon T]$,*

$$\mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\frac{\mathbb{1}(A^* = a)}{p^{\text{TS-rnd}}_{t,a}}\,\middle|\,\mathcal{H}_t\right] \le 1 + 2\epsilon_{\text{rnd}}.$$

Proof

$$\mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\frac{\mathbb{1}(A^*=a)}{p_{t,a}^{\text{TS-rnd}}}\,\middle|\,\mathcal{H}_t\right] = \mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\frac{\mathbb{1}(A^*=a)}{p_{t,a}^*}\frac{p_{t,a}^*}{p_{t,a}^{\text{TS-rnd}}}\,\middle|\,\mathcal{H}_t\right]$$
$$\underbrace{=}_{(a)}\mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\frac{\mathbb{1}(A^*=a)}{p_{t,a}^*}\frac{p_{t,a}^{\text{TS}}}{p_{t,a}^{\text{TS-rnd}}}\,\middle|\,\mathcal{H}_t\right]\underbrace{\leq}_{(b)}\frac{1}{1-\epsilon_{\text{rnd}}}\mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\frac{\mathbb{1}(A^*=a)}{p_{t,a}^*}\,\middle|\,\mathcal{H}_t\right]$$
$$=\frac{1}{1-\epsilon_{\text{rnd}}}\cdot 1\underbrace{\leq}_{(c)}1+2\epsilon_{\text{rnd}}$$

- Above, equality (a) holds because $p_{t,a}^* = \mathbb{P}(A^*=a \mid \mathcal{H}_t) = \mathbb{P}(A_u = a \mid \mathcal{H}_t) = p_{t,a}^{\text{TS}}$.
- Inequality (b) holds by rounding Assumption 4.
- Inequality (c) holds since $\epsilon_{\text{rnd}} \leq 0.5$ by Assumption 4 and since by Taylor Series expansion for any ϵ with $0<\epsilon<1$, $\frac{1}{1-\epsilon} = \sum_{k=0}^{\infty}\epsilon^k = 1+\epsilon+\sum_{k=2}^{\infty}\epsilon^k \leq 1+2\epsilon$.

■

Lemma 10 (Simplifying Value of Progressive Feedback Term)**.** *Under Assumptions 1 and 3,*

$$\text{VoPF}(t,z) = \frac{1}{2}\log\frac{\text{Var}\big(\bar{R}_a \mid Z_a=z, \{Y_u(a)\}_{u\in\mathcal{U}_{<(t-d_{\max})}}\big)}{\text{Var}\big(\bar{R}_a \mid Z_a=z, \{\tilde{Y}_{u,t}(a)\}_{u\in\mathcal{U}_{<t}}\big)}. \tag{24}$$

Proof Note, by definition of VoPF from (11),

$$\text{VoPF}(t,z) = I\left(\bar{R}_a;\, \left\{\tilde{Y}_{u,t}(a)\right\}_{u\in\mathcal{U}_{<t}} \mid Z_a=z, \left\{Y_u(a)\right\}_{u\in\mathcal{U}_{<(t-d_{\max})}}\right)$$
$$\underbrace{=}_{(a)}H\left(\bar{R}_a \mid Z_a=z, \left\{Y_u(a)\right\}_{u\in\mathcal{U}_{<(t-d_{\max})}}\right) - H\left(\bar{R}_a \mid Z_a=z, \left\{\tilde{Y}_{u,t}(a)\right\}_{u\in\mathcal{U}_{<t}}\right)$$

$$\underbrace{=}_{(b)}\mathbf{E}\left[\frac{1}{2}\log\left\{2\pi e\text{Var}\left(\bar{R}_a \mid Z_a=z, \left\{Y_u(a)\right\}_{u\in\mathcal{U}_{<(t-d_{\max})}}\right)\right\}\right]$$
$$-\mathbf{E}\left[\frac{1}{2}\log\left[2\pi e\text{Var}\left(\bar{R}_a \mid Z_a=z, \left\{\tilde{Y}_{u,t-1}(a)\right\}_{u\in\mathcal{U}_{<t}}\right)\right]\right]$$
$$=\frac{1}{2}\mathbf{E}\left[\log\frac{\text{Var}\big(\bar{R}_a \mid Z_a=z, \{Y_{u,t}(a)\}_{u\in\mathcal{U}_{<(t-d_{\max})}}\big)}{\text{Var}\left(\bar{R}_a \mid Z_a=z, \{\tilde{Y}_{u,t}(a)\}_{u\in\mathcal{U}_{<t}}\right)}\,\middle|\,Z_a=z\right]$$
$$\underbrace{=}_{(c)}\frac{1}{2}\log\frac{\text{Var}\big(\bar{R}_a \mid Z_a=z, \{Y_{u,t}(a)\}_{u\in\mathcal{U}_{<(t-d_{\max})}}\big)}{\text{Var}\left(\bar{R}_a \mid Z_a=z, \{\tilde{Y}_{u,t}(a)\}_{u\in\mathcal{U}_{<t}}\right)}$$

- Equality (a): This holds by standard results in information theory. Note as is standard in information theory, the mutual information $I\left(\bar{R}_a;\, \{\tilde{Y}_{u,t}(a)\}_{u\in\mathcal{U}_{<t}} \mid Z_a = z, \{Y_u(a)\}_{u\in\mathcal{U}_{<(t-d_{\max})}}\right)$ is a constant as it marginalizes over $\{Y_u(a)\}_{u\in\mathcal{U}_{<(t-d_{\max})}}$.
  Similarly, $H\left(\bar{R}_a \mid Z_a = z, \{Y_u(a)\}_{u\in\mathcal{U}_{<(t-d_{\max})}}\right)$ denotes the conditional entropy and is marginalizes over $\{Y_u(a)\}_{u\in\mathcal{U}_{<(t-d_{\max})}}$.
- Equality (b): This holds by the formula for entropy of a Gaussian random variable.
- Equality (c): Under our Gaussian assumption, the variance terms $\mathrm{Var}\big(\bar{R}_a \mid Z_a = z, \{Y_{u,t}(a)\}_{u\in\mathcal{U}_{<(t-d_{\max})}}\big)$ and $\mathrm{Var}\left(\bar{R}_a \mid Z_a = z, \{\tilde{Y}_{u,t}(a)\}_{u\in\mathcal{U}_{<t}}\right)$ are constants.

■

Lemma 11 (Simplifying Mean Reward Variance Given Progressive Feedback)**.** *Under Assumptions 1-3,*

$$\mathbf{E}\left[\mathrm{Var}\left(\bar{R}_a \mid Z_a, \{\tilde{Y}_{u,t}(a)\}_{u\in\mathcal{U}_{<t}}\right)\right] = \mathbf{E}\left[\frac{\sigma_R^2(Z_a)\cdot\exp\big(-2\cdot\mathrm{VoPF}(t,Z_a)\big)}{\sigma_R^2(Z_a)\cdot(r_1^\top\Sigma_{Z_a}r_1)^{-1}+|\mathcal{U}_{<(t-d_{\max})}|}\right]$$

Proof For any $a\in\mathcal{A}$,

$$\begin{aligned}
&\mathbf{E}\left[\mathrm{Var}\left(\bar{R}_a \mid Z_a, \{\tilde{Y}_{u,t}(a)\}_{u\in\mathcal{U}_{<t}}\right)\right]\\
&= \mathbf{E}\left[\mathrm{Var}\left(\bar{R}_a \mid Z_a, \{Y_{u,t}(a)\}_{u\in\mathcal{U}_{<(t-d_{\max})}}\right)\frac{\mathrm{Var}\left(\bar{R}_a \mid Z_a, \{\tilde{Y}_{u,t}(a)\}_{u\in\mathcal{U}_{<t}}\right)}{\mathrm{Var}\big(\bar{R}_a \mid Z_a, \{Y_{u,t}(a)\}_{u\in\mathcal{U}_{<(t-d_{\max})}}\big)}\right]\\
&\underbrace{=}_{(i)} \mathbf{E}\left[\frac{\sigma_R^2(Z_a)}{\sigma_R^2(Z_a)\cdot(r_1^\top\Sigma_{Z_a}r_1)^{-1}+|\mathcal{U}_{<(t-d_{\max})}|}\cdot\frac{\mathrm{Var}\left(\bar{R}_a \mid Z_a, \{\tilde{Y}_{u,t}(a)\}_{u\in\mathcal{U}_{<t}}\right)}{\mathrm{Var}\big(\bar{R}_a \mid Z_a, \{Y_{u,t}(a)\}_{u\in\mathcal{U}_{<(t-d_{\max})}}\big)}\right]\\
&\underbrace{=}_{(ii)} \mathbf{E}\left[\frac{\sigma_R^2(Z_a)\cdot\exp\big(-2\cdot\mathrm{VoPF}(t,Z_a)\big)}{\sigma_R^2(Z_a)\cdot(r_1^\top\Sigma_{Z_a}r_1)^{-1}+|\mathcal{U}_{<(t-d_{\max})}|}\right]
\end{aligned}$$

Equality (i) holds since with probability 1,

$$\mathrm{Var}\left(\bar{R}_a \mid Z_a, \{Y_{u,t}(a)\}_{u\in\mathcal{U}_{<(t-d_{\max})}}\right) = \frac{\sigma_R^2(Z_a)}{\sigma_R^2(Z_a)\cdot(r_1^\top\Sigma_{1,Z_a}r_1)^{-1}+|\mathcal{U}_{<(t-d_{\max})}|}. \tag{25}$$

The equality above holds by applying the formula for updating a scalar Gaussian random variable $\bar{R}_a$ based on i.i.d noisy measurements $\{R(Y_u)\}_{u\in\mathcal{U}_{<t}}$; recall that we use the notation $\sigma_R^2(z) = \mathrm{Var}\big(R(Y_u(a)) \mid \bar{R}_a, Z_a = z\big)$. Equality (ii) above holds by Lemma 10. ■

D.3 Completing the proof of Theorem 1

Theorem 1 (Regret with Progressive Feedback). *Let Assumptions 1-4 hold, and let $\Sigma_{1,z}$ and V_z be invertible for all $z \in \mathcal{Z}$. Then under Impatient Bandit Thompson Sampling algorithm $\pi^{\text{TS-rnd}}$ (Algorithm 2), for any $t > 1$,*

$$\mathbf{E}\left[\Delta_t(\pi^{\text{TS-rnd}})\right] \leq \sqrt{\mathbf{E}\left[\frac{\exp\big(-2\cdot \text{VoPF}(t, Z_a)\big)\cdot \sigma_R^2(Z_a)}{\sigma_R^2(Z_a)\cdot (r_1^\top \Sigma_{Z_a} r_1)^{-1} + |\mathcal{U}_{<(t-d_{\max})}|}\right]} \times \sqrt{2|\mathcal{A}|\log(|\mathcal{A}|)} + O(\epsilon_{\text{rnd}}).$$

Proof Recall that, by Lemma 7, that

$$\mathbf{E}\left[\Delta_t(\pi^{\text{TS-rnd}})\right] \leq \sqrt{2\log(|\mathcal{A}|)\mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\sum_{a\in\mathcal{A}} p^*_{t,a}\sigma^2_{t,a}\right]} + O(\epsilon_{\text{rnd}})$$

where $p^*_{t,a} = \mathbb{P}(A^* = a \mid \mathcal{H}_t)$. By the above result, it is sufficient for the Theorem to show

$$\mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\sum_{a\in\mathcal{A}} p^*_{t,a}\sigma^2_{t,a}\right] \leq (1+2\epsilon_{\text{rnd}})|\mathcal{A}|\cdot \mathbf{E}\left[\frac{\sigma_R^2(Z_a)\cdot\exp\big(-2\cdot\text{VoPF}(t,Z_a)\big)}{\sigma_R^2(Z_a)\cdot(r_1^\top\Sigma_{Z_a}r_1)^{-1} + |\mathcal{U}_{<(t-d_{\max})}|}\right]. \quad (26)$$

The remainder of the proof will be showing (26) holds. Note that

$$\begin{aligned}
&\mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\sum_{a\in\mathcal{A}} p^*_{t,a}\sigma^2_{t,a}\right] = \mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\sum_{a\in\mathcal{A}} \mathbb{1}(A^*=a)\sigma^2_{t,a}\right] \\
&= \mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\sum_{a\in\mathcal{A}} \mathbb{1}(A^*=a) r_1^\top \text{Cov}\left(\theta_a \mid \mathcal{H}_t\right) r_1\right] = \sum_{a\in\mathcal{A}} r_1^\top \mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\mathbb{1}(A^*=a)\text{Cov}\left(\theta_a \mid \mathcal{H}_t\right)\right] r_1 \\
&\leq \sum_{a\in\mathcal{A}} r_1^\top \mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\mathbb{1}(A^*=a) C^{\text{Full}}_{t,a}\left(\Sigma^{-1}_{1,Z_a} + m\sum_{t'=1}^{t-1}\frac{\sum_{j=1}^J \mathbb{1}\{t > t'+d_j\}\ell^{(j)}_{Z_a}(\ell^{(j)}_{Z_a})^\top}{p^{\text{TS-rnd}}_{t',a}}\right) C^{\text{Full}}_{t,a}\right] r_1
\end{aligned}$$

The last inequality above holds by Lemma 8. Additionally, by (22), $C^{\text{Full}}_{t,a}$ is non-random given $\{Z_a\}_{a\in\mathcal{A}}$. Thus, by the law of iterated expectations, the above equals the following:

$$= \sum_{a\in\mathcal{A}} r_1^\top \mathbf{E}\left[C^{\text{Full}}_{t,a}\mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\mathbb{1}(A^*=a)\left(\Sigma^{-1}_{1,Z_a} + m\sum_{t'=1}^{t-1}\frac{\sum_{j=1}^J \mathbb{1}\{t > t'+d_j\}\ell^{(j)}_{Z_a}(\ell^{(j)}_{Z_a})^\top}{p^{\text{TS-rnd}}_{t',a}}\right)\middle| \{Z_a\}_{a\in\mathcal{A}}\right] C^{\text{Full}}_{t,a}\right] r_1$$

By moving terms that are constant given $\{Z_a\}_{a\in\mathcal{A}}$ out of the expectation, we get that

$$= \sum_{a\in\mathcal{A}} r_1^\top \mathbf{E}\left[C^{\text{Full}}_{t,a}\left(\Sigma^{-1}_{1,Z_a} + m\sum_{t'=1}^{t-1}\mathbf{E}_{\pi^{\text{TS-rnd}}}\left[\frac{\mathbb{1}(A^*=a)}{p^{\text{TS-rnd}}_{t',a}}\middle|\{Z_a\}_{a\in\mathcal{A}}\right]\sum_{j=1}^J \mathbb{1}\{t>t'+d_j\}\ell^{(j)}_{Z_a}(\ell^{(j)}_{Z_a})^\top\right) C^{\text{Full}}_{t,a}\right] r_1$$

By Lemma 9 (recall that $\{Z_a\}_{a\in\mathcal{A}}$ is part of the history $\mathcal{H}_t$),

$$\leq (1+2\epsilon_{\text{rnd}})\sum_{a\in\mathcal{A}} r_1^\top \mathbf{E}\left[C^{\text{Full}}_{t,a}\left(\Sigma^{-1}_{1,Z_a} + m\sum_{t'=1}^{t-1}\sum_{j=1}^J \mathbb{1}\{t>t'+d_j\}\ell^{(j)}_{Z_a}(\ell^{(j)}_{Z_a})^\top\right) C^{\text{Full}}_{t,a}\right] r_1$$

By the definition of $C_{t,a}^{\text{Full}}$ from (22),

$$= (1+2\epsilon_{\text{rnd}}) \sum_{a\in\mathcal{A}} r_1^\top \mathbf{E}\left[C_{t,a}^{\text{Full}}\right] r_1 = (1+2\epsilon_{\text{rnd}}) \sum_{a\in\mathcal{A}} \mathbf{E}\left[r_1^\top \text{Cov}\left(\theta_a \mid Z_a, \{\tilde{Y}_{u,t}(a)\}_{u\in\mathcal{U}_{<t}}\right) r_1\right]$$

$$\underbrace{=}_{(i)} (1+2\epsilon_{\text{rnd}})|\mathcal{A}|\mathbf{E}\left[\text{Var}\left(\bar{R}_a \mid Z_a, \{\tilde{Y}_{u,t}(a)\}_{u\in\mathcal{U}_{<t}}\right)\right]$$

$$\underbrace{=}_{(ii)} (1+2\epsilon_{\text{rnd}})|\mathcal{A}|\mathbf{E}\left[\frac{\sigma_R^2(Z_a)\cdot\exp\big(-2\cdot\text{VoPF}(t,Z_a)\big)}{\sigma_R^2(Z_a)\cdot(r_1^\top\Sigma_{Z_a}r_1)^{-1}+|\mathcal{U}_{<(t-d_{\max})}|}\right].$$

Equality (i) above holds for any action $a\in\mathcal{A}$, since by Assumption 2, $a\in\mathcal{A}$ are i.i.d. from the same distribution. Finally, equality (ii) holds by Lemma 11. ■

D.4 Proof of Corollary 4

Corollary 4 (Regret of Impatient Bandit Algorithm without Action Features Z_a)**.** *Let Assumptions 1-4 hold, and let Σ_1 and V be invertible. Then under the Impatient Bandit Thompson Sampling algorithm $\pi^{\text{TS-rnd}}$ (Algorithm 2), for any $t>1$,*

$$\mathbf{E}\left[\Delta_t(\pi^{\text{TS-rnd}})\right] \le \underbrace{\exp\big[-\text{VoPF}(t)\big]}_{\substack{\text{Decrease from}\\ \text{Progressive Feedback}}} \cdot \underbrace{\sigma_R\sqrt{\frac{2|\mathcal{A}|\log|\mathcal{A}|}{\sigma_R^2(r_1^\top\Sigma_1 r_1)^{-1}+|\mathcal{U}_{<(t-d_{\max})}|}}}_{\text{Regret under Delayed Rewards}} + \underbrace{O(\epsilon_{\text{rnd}})}_{\text{Rounding error}}.$$

Proof In the case in which there are no action features Z_a, recall from (7) that $\text{VoPF}(t) = I\big(\bar{R}_a;\ \{\tilde{Y}_{u,t}(a): u\in\mathcal{U}_{<t}\} \mid \{Y_u(a): u\in\mathcal{U}_{<(t-d_{\max})}\}\big)$. By Theorem 1, we have that

$$\mathbf{E}\left[\Delta_t(\pi^{\text{TS-rnd}})\right] \le \sqrt{\mathbf{E}\left[\exp\big(-2\cdot\text{VoPF}(t)\big)\cdot\sigma_R^2(Z_a)\right]} \times \sqrt{\frac{2|\mathcal{A}|\log(|\mathcal{A}|)}{|\mathcal{U}_{<(t-d_{\max})}|}} + O(\epsilon_{\text{rnd}})$$

$$= \sqrt{\exp\big(-2\cdot\text{VoPF}(t)\big)\cdot\sigma_R^2} \times \sqrt{\frac{2|\mathcal{A}|\log(|\mathcal{A}|)}{|\mathcal{U}_{<(t-d_{\max})}|}} + O(\epsilon_{\text{rnd}}).$$

Note for last equality above we use the fact when there are no action features Z_a, $\sigma_R^2 = \sigma_R^2(Z_a)$ is a constant and so is $\text{VoPF}(t)$. ■

Proposition 5 (Regret Under Extremes)**.** *Let the conditions of Corollary 4 hold.*
(1) Completely Uninformative Intermediate Feedback (Delayed Rewards). *In the case that the intermediate feedback is completely uninformative, i.e.,* $\text{VoPF}\big(t\big)=0$ *for all t,*

$$\mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T}\Delta_t(\pi^{\text{TS-rnd}})\right] \ \le\ \frac{d_{\max}}{T}\sqrt{2(r_1^\top\Sigma_1 r_1)\log(|\mathcal{A}|)} + 3\sigma_R\sqrt{\frac{2|\mathcal{A}|\log(|\mathcal{A}|)}{Tm}} + O(\epsilon_{\text{rnd}}).$$

(2) Perfect Intermediate Feedback (No Delay). *If we have a perfect surrogate, i.e.,* $\mathrm{Corr}\big(Y_u^{(1)}, R(Y_u)\big) = 1$ *and* $d_1 = 0$,

$$\mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T}\Delta_t(\pi^{\text{TS-rnd}})\right] \le 3\sigma_R\sqrt{\frac{2|\mathcal{A}|\log(|\mathcal{A}|)}{Tm}} + O(\epsilon_{\text{rnd}}).$$

Proof
(1) Completely Uninformative Intermediate Feedback (Delayed Rewards). We first bound the regret under the first $d_{\max}$ batches. Let π_{uniform} denote the policy that randomly selects an action uniformly from $\mathcal{A}$ to play exclusively for the first $d_{\max}$ batches:

$$\begin{aligned}\mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{d_{\max}}\Delta_t(\pi^{\text{TS-rnd}})\right] &\le \mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{d_{\max}}\Delta_t(\pi_{\text{uniform}})\right] \underbrace{\le}_{(a)} \frac{d_{\max}}{T}\left(\mathbf{E}\left[\operatorname*{argmax}_{a\in\mathcal{A}}\left\{r_1^\top\theta_a\right\}\right] - r_1^\top\mu_1\right)\\ &= \frac{d_{\max}}{T}\mathbf{E}\left[\operatorname*{argmax}_{a\in\mathcal{A}}\left\{r_1^\top(\theta_a-\mu_1)\right\}\right] \underbrace{\le}_{(b)} \frac{d_{\max}}{T}\sqrt{2(r_1^\top\Sigma_1 r_1)\log(|\mathcal{A}|)} \quad (27)\end{aligned}$$

Inequality (a) above uses that the reward function $R(Y_u(a)) = r_1^\top Y_u(a) + r_2$ is a linear function, which means $\mathbf{E}[R(Y_u(a)) \mid \theta_a] = r_1^\top\theta_a$. Recall $\theta_a \sim N(\mu_1, \Sigma_1)$ according to Lemma 2. Inequality (b) uses that $\mathbf{E}\left[\operatorname{argmax}_{a\in\mathcal{A}}\left\{r_1^\top(\theta_a-\mu_1)\right\}\right] = \mathbf{E}\left[\operatorname{argmax}_{a\in\mathcal{A}} W_a\right]$ for $W_a \overset{\text{i.i.d.}}{\sim} N(0, r_1^\top\Sigma_1 r_1)$ over $a \in \mathcal{A}$ and basic inequality bounds on the expectation of the maxima of independent Gaussian random variables.

We now consider the subsequent batches (after the first $d_{\max}$ batches). By Corollary 4,

$$\mathbf{E}\left[\frac{1}{T}\sum_{t=d_{\max}+1}^{T}\Delta_t(\pi^{\text{TS-rnd}})\right] \le \sigma_R\sqrt{2|\mathcal{A}|\log(|\mathcal{A}|)}\cdot\frac{1}{T}\sum_{t=d_{\max}+1}^{T}\sqrt{\frac{\exp\big(-2\cdot\mathrm{VoPF}\big(t\big)\big)}{\sigma_R^2\cdot(r_1^\top\Sigma_1 r_1)^{-1} + |\mathcal{U}_{<(t-d_{\max})}|}} + O(\epsilon_{\text{rnd}}). \quad (28)$$

Note when $\mathrm{VoPF}\big(t\big) = 0$ for all t,

$$\begin{aligned}\frac{1}{T}\sum_{t=d_{\max}+1}^{T}\sqrt{\frac{\exp\big(-2\cdot\mathrm{VoPF}\big(t\big)\big)}{\sigma_R^2(r_1^\top\Sigma_1 r_1)^{-1} + |\mathcal{U}_{<t-d_{\max}}|}} &\le \frac{1}{T}\sum_{t=d_{\max}+1}^{T}\frac{1}{\sqrt{|\mathcal{U}_{<t-d_{\max}}|}}\\ &\underbrace{=}_{(a)} \frac{1}{T\sqrt{m}}\sum_{t=d_{\max}+1}^{T}\frac{1}{\sqrt{t-d_{\max}}} = \frac{1}{T\sqrt{m}}\sum_{t=1}^{T-d_{\max}}\frac{1}{\sqrt{t}}\\ &\underbrace{\le}_{(b)} \frac{2\sqrt{T-d_{\max}}+1}{T\sqrt{m}} = \frac{2}{\sqrt{Tm}} + \frac{1}{T\sqrt{m}} \le \frac{3}{\sqrt{Tm}}\end{aligned}$$

For equality (a) above, we use that $|\mathcal{U}_{<t-d_{\max}}| = m\cdot\max(0, t-d_{\max})$. Equality (b) holds since $\frac{1}{\sqrt{t}}$ is decreasing in t, $\sum_{t=1}^{T}\frac{1}{\sqrt{T}} \le \int_0^T\frac{1}{\sqrt{t}}dt = 2\sqrt{T}$. By the above result, (27), and (28), we have shown (9) holds.

(2) Perfect Intermediate Feedback (No Delay). By Corollary 4,

$$\mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T}\Delta_t(\pi^{\text{TS-rnd}})\right] \le \sigma_R\sqrt{2|\mathcal{A}|\log(|\mathcal{A}|)}\cdot\frac{1}{T}\sum_{t=1}^{T}\sqrt{\frac{\exp\big(-2\cdot\text{VoPF}\big(t\big)\big)}{\sigma_R^2\cdot(r_1^\top\Sigma_1 r_1)^{-1}+|\mathcal{U}_{<(t-d_{\max})}|}}+O(\epsilon_{\text{rnd}})$$

$$\underbrace{=}_{(a)}\sigma_R\sqrt{2|\mathcal{A}|\log(|\mathcal{A}|)}\cdot\frac{1}{T}\sum_{t=1}^{T}\sqrt{\text{Var}\big(\bar{R}_a\mid Z_a=z,\{\tilde{Y}_{u,t}(a)\}_{u\in\mathcal{U}_{<t}}\big)}+O(\epsilon_{\text{rnd}})$$

$$\underbrace{=}_{(b)}\sigma_R\sqrt{2|\mathcal{A}|\log(|\mathcal{A}|)}\cdot\frac{1}{T}\sum_{t=1}^{T}\sqrt{\frac{1}{\sigma_R^2\cdot(r_1^\top\Sigma_1 r_1)^{-1}+|\mathcal{U}_{<t}|}}+O(\epsilon_{\text{rnd}})$$

$$\underbrace{\le}_{(c)}\sigma_R\sqrt{2|\mathcal{A}|\log(|\mathcal{A}|)}\cdot\frac{1}{T\sqrt{m}}\sum_{t=1}^{T}\sqrt{\frac{1}{t}}+O(\epsilon_{\text{rnd}})$$

$$\underbrace{\le}_{(d)}\sigma_R\sqrt{2|\mathcal{A}|\log(|\mathcal{A}|)}\cdot\frac{2\sqrt{T}+1}{T\sqrt{m}}+O(\epsilon_{\text{rnd}})\le\sigma_R\sqrt{2|\mathcal{A}|\log(|\mathcal{A}|)}\cdot\frac{3}{\sqrt{Tm}}+O(\epsilon_{\text{rnd}})$$

Above (a) holds since $\text{Var}(\bar{R}_a\mid Z_a=z,\{Y_{u,t}(a)\}_{u\in\mathcal{U}_{<(t-d_{\max})}})=\{(r_1^\top\Sigma_1 r_1)^{-1}+|\mathcal{U}_{<(t-d_{\max})}|\}^{-1}$ and that by Lemma 10,

$$\exp\big(-2\cdot\text{VoPF}\big(t\big)\big)=\frac{\text{Var}\big(\bar{R}_a\mid Z_a=z,\{\tilde{Y}_{u,t}(a)\}_{u\in\mathcal{U}_{<t}}\big)}{\text{Var}\big(\bar{R}_a\mid Z_a=z,\{Y_{u,t}(a)\}_{u\in\mathcal{U}_{<(t-d_{\max})}}\big)}$$

Equality (b) uses the formula for the posterior variance of a Gaussian, as well as the fact that we have a perfect surrogate, i.e., that $\text{Corr}(R_u,Y_u^{(1)})=1$. For inequality (c) above, we use that $|\mathcal{U}_{\le t}|=m\cdot t$. Inequality (d) holds since $\frac{1}{\sqrt{t}}$ is decreasing in t, $\sum_{t=1}^{T}\frac{1}{\sqrt{T}}\le\int_0^T\frac{1}{\sqrt{t}}dt=2\sqrt{T}$. Thus we have shown that (10) holds. ■

D.5 Derivation of Remark 3

By Lemma 10, under Assumptions 1 and 3,

$$\text{VoPF}(t)=\frac{1}{2}\log\frac{\text{Var}\big(\bar{R}_a\mid\{Y_u(a)\}_{u\in\mathcal{U}_{<(t-d_{\max})}}\big)}{\text{Var}\big(\bar{R}_a\mid\{\tilde{Y}_{u,t}(a)\}_{u\in\mathcal{U}_{<t}}\big)}. \tag{29}$$

Recall that $\theta_a\sim N(\mu_1,\Sigma_1)$ and $Y_u(a)\mid\theta_a\sim N(\theta_a,V)$, where

$$\Sigma_1=\begin{bmatrix}1&\rho\\\rho&1\end{bmatrix}\qquad\text{and}\qquad V=\sigma_R^2\begin{bmatrix}1&\rho\\\rho&1\end{bmatrix}.$$

Numerator of (29) Note that the posterior distribution $\theta_a\mid\{Y_{u,t}(a)\}_{u\in\mathcal{U}_{<(t-d_{\max})}}$ is multivariate Gaussian with the following posterior covariance matrix:

$$C\triangleq(\Sigma_1^{-1}+m(t-d_{\max})V^{-1})^{-1}=(1+m(t-d_{\max})\sigma_R^{-2})^{-1}\Sigma_1.$$

Above, recall we use m to refer to the number of users per batch. Thus, the posterior variance of the expected reward $\bar{R}_a$ is

$$\mathbf{Var}[\bar{R}_a\mid\{Y_{u,t}(a)\}_{u\in\mathcal{U}_{<(t-d_{\max})}}]=(1+m(t-d_{\max})\sigma_R^{-2})^{-1}$$

Denominator of (29) Now note that the posterior distribution $\theta_a \mid \{\tilde{Y}_{u,t}(a)\}_{u\in\mathcal{U}_{<t}}$ is multivariate Gaussian with the following posterior covariance matrix:

$$\begin{aligned} &C - ((1+m(t-d_{\max})\sigma_R^{-2})^{-1} + \sigma_R^2)^{-1}\boldsymbol{c}\boldsymbol{c}^\top \\ &\quad = (1+n\sigma_R^{-2})^{-1}\left(\begin{bmatrix}1 & \rho\\ \rho & 1\end{bmatrix} - (1+\sigma_R^2+m(t-d_{\max}))^{-1}\begin{bmatrix}1 & \rho\\ \rho & \rho^2\end{bmatrix}\right) \end{aligned}$$

where $\boldsymbol{c} = (1+m(t-d_{\max})\sigma_R^2)^{-1}\begin{bmatrix}1 & \rho\end{bmatrix}$ is the first row of C. As a result,

$$\begin{aligned} \mathbf{Var}[\bar{R}_a \mid \{\tilde{Y}_{u,t}(a)\}_{u\in\mathcal{U}_{<t}}] &= (1+m(t-d_{\max})\sigma_R^{-2})^{-1}\left(1 - \frac{\rho^2}{\sigma_R^2+m(t-d_{\max})+1}\right) \\ &= (1+m(t-d_{\max})\sigma_R^{-2})^{-1}\left(\frac{\sigma_r^2+m(t-d_{\max})+1-\rho^2}{\sigma_R^2+m(t-d_{\max})+1}\right) \end{aligned}$$

Thus, we can rewrite (29) as follows:

$$\mathrm{VoPF}(t) = \frac{1}{2}\log\left(\frac{\sigma_R^2+m(t-d_{\max})+1}{\sigma_r^2+m(t-d_{\max})+1-\rho^2}\right) \quad \forall t \geq d_{\max}.$$

Appendix E. Rounding Procedure

Algorithm 6 Iterative Decrement Rounding Algorithm

Inputs: Probabilities $\{p_{t,a}^{\mathrm{TS}}\}_{a\in\mathcal{A}}$, Granularity m
for $a \in \mathcal{A}$ **do**
  $p_{t,a}^{\text{TS-rnd}} \leftarrow \frac{\lceil m p_{t,a}^{\mathrm{TS}}\rceil}{m}$
while $\sum_{a\in\mathcal{A}} p_{t,a}^{\text{TS-rnd}} > 1$ **do**
  $\alpha \leftarrow \mathrm{argmax}_{a\in\mathcal{A}} \frac{p_{t,a}^{\text{TS-rnd}} - 1/m}{p_{t,a}^{\mathrm{TS}}}$
  $p_{t,\alpha}^{\text{TS-rnd}} \leftarrow p_{t,\alpha}^{\text{TS-rnd}} - \frac{1}{m}$
Output: $\{p_{t,a}^{\text{TS-rnd}}\}_{a\in\mathcal{A}}$

Proposition 12 (Iterative Decrement Rounding Procedure)**.** *Rounding procedure Algorithm 6 satisfies Assumption 4 holds as long as $|\mathcal{A}| \geq 2$ and $m \geq 2|\mathcal{A}|$. Specifically, for any value of m, Assumption 4 will hold for $\epsilon_{\mathrm{rnd}} = \frac{|\mathcal{A}|}{m}$.*

Proof Let $q_a^{(0)} \triangleq \frac{\lceil m p_{t,a}^{\mathrm{TS}}\rceil}{m}$ denote the initially rounded-up probabilities. Since $p_{t,a}^{\mathrm{TS}} \leq q_a^{(0)} < p_{t,a}^{\mathrm{TS}} + \frac{1}{m}$, we have that

$$1 \leq \sum_{a\in\mathcal{A}} q_a^{(0)} < 1 + \frac{|\mathcal{A}|}{m}.$$

Moreover, $\sum_{a\in\mathcal{A}} q_a^{(0)}$ lies on the grid $\{j/m : j \in \mathbb{Z}\}$. Therefore the excess mass $\sum_{a\in\mathcal{A}} q_a^{(0)} - 1$ is an integer multiple of $1/m$ and is strictly less than $|\mathcal{A}|/m$. Hence Algorithm 6 performs at most $|\mathcal{A}|$ decrement steps, and after these steps the output $p_t^{\text{TS-rnd}}$ satisfies $\sum_{a\in\mathcal{A}} p_{t,a}^{\text{TS-rnd}} = 1$.

Absolute Error Bound. For each a, $0 \le q_a^{(0)} - p_{t,a}^{\text{TS}} < \frac{1}{m}$. The algorithm only decreases the initial rounded up value $q_a^{(0)}$, and the total number of decrements is at most $|\mathcal{A}|$. Therefore no coordinate can move downward by more than $|\mathcal{A}|/m$, which implies

$$\left|p_{t,a}^{\text{TS-rnd}} - p_{t,a}^{\text{TS}}\right| \le \frac{|\mathcal{A}|}{m}.$$

Relative Error Bound. If $q_a^{(0)}$ is never decremented, then $p_{t,a}^{\text{TS-rnd}} = q_a^{(0)} \ge p_{t,a}^{\text{TS}}$, so $\frac{p_{t,a}^{\text{TS-rnd}}}{p_{t,a}^{\text{TS}}} \ge 1$.

Now suppose coordinate a is decremented at least once, and consider the final time at which a is decremented. Let $\tilde{q} = (\tilde{q}_a, \ldots, \tilde{q}_{|\mathcal{A}|})$ denote the vector immediately before this decrement. By the definition of the decrement rule in line 5 of Algorithm 6,

$$a \in \operatorname*{argmax}_{\alpha \in \mathcal{A}} \frac{\tilde{q}_\alpha - 1/m}{p_{t,\alpha}^{\text{TS}}}.$$

Immediately after this decrement, and since this is the final decrement of coordinate a, $\frac{p_{t,a}^{\text{TS-rnd}}}{p_{t,a}^{\text{TS}}} = \frac{\tilde{q}_a - 1/m}{p_{t,a}^{\text{TS}}}$. Thus,

$$\frac{p_{t,a}^{\text{TS-rnd}}}{p_{t,a}^{\text{TS}}} \ge \max_{\alpha \in \mathcal{A}} \frac{\tilde{q}_\alpha - 1/m}{p_{t,\alpha}^{\text{TS}}}.$$

Since the algorithm only decreases $\tilde{q}_\alpha$ as it runs, the final rounded probability satisfies $p_{t,\alpha}^{\text{TS-rnd}} \le \tilde{q}_\alpha$, and therefore

$$\frac{p_{t,a}^{\text{TS-rnd}}}{p_{t,a}^{\text{TS}}} \ge \frac{p_{t,\alpha}^{\text{TS-rnd}} - 1/m}{p_{t,\alpha}^{\text{TS}}} \qquad \text{for all } \alpha \in \mathcal{A}. \tag{30}$$

Thus, we have that

$$\frac{p_{t,a}^{\text{TS-rnd}}}{p_{t,a}^{\text{TS}}} = \frac{p_{t,a}^{\text{TS-rnd}}}{p_{t,a}^{\text{TS}}} \sum_{\alpha \in \mathcal{A}} p_{t,\alpha}^{\text{TS}} \underbrace{\ge}_{(i)} \sum_{\alpha \in \mathcal{A}} \left(p_{t,\alpha}^{\text{TS-rnd}} - \frac{1}{m} \right) = 1 - \frac{|\mathcal{A}|}{m}.$$

Inequality (i) above holds by (30) by multiplying both sides by $p_{t,\alpha}^{\text{TS-rnd}}$. ■

References

Sas/stat 15.1 user's guide the mixed procedure, 2018. URL `https://documentation.sas.com/api/collections/pgmsascdc/9.4_3.4/docsets/statug/content/mixed.pdf?locale=en#nameddest=statug_mixed_details01`.

S. Agrawal and N. Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International conference on machine learning*, pages 127–135. PMLR, 2013.

D. J. Aldous. Exchangeability and related topics. In *École d'Été de Probabilités de Saint-Flour XIII—1983*, pages 1–198. Springer, 2006.

A. Anderer, H. Bastani, and J. Silberholz. Adaptive clinical trial designs with surrogates: When should we bother? *Management Science*, 68(3):1982–2002, 2022.

S. Athey, R. Chetty, G. Imbens, and H. Kang. Estimating treatment effects using multiple surrogates: The role of the surrogate score and the surrogate index. 2016.

E. M. Azevedo, A. Deng, J. L. Montiel Olea, and E. G. Weyl. Empirical bayes estimation of treatment effects with many a/b tests: An overview. In *AEA Papers and Proceedings*, 2019.

E. M. Azevedo, A. Deng, J. L. Montiel Olea, J. Rao, and E. G. Weyl. A/b testing with fat tails. *Journal of Political Economy*, 128(12):4614–000, 2020.

B. Baran, G. D. Junior, A. Danylenko, O. S. Folorunso, G. Forsum, M. Lefarov, L. Maystre, and Y. Zhao. Accelerating creator audience building through centralized exploration. In *Proceedings of RecSys 2023*, Singapore, Sept. 2023.

I. Bistritz, Z. Zhou, X. Chen, N. Bambos, and J. Blanchet. Online exp3 learning in adversarial bandits with delayed feedback. In *Advances in Neural Information Processing Systems*, 2019.

V. Bogina and T. Kuflik. Incorporating dwell time in session-based recommendations with recurrent neural networks. In *RecTemp@ RecSys*, pages 57–59, 2017.

G. Casella. An introduction to empirical bayes data analysis. *The American Statistician*, 1985.

D. Cheng, A. N. Ananthakrishnan, and T. Cai. Robust and efficient semi-supervised estimation of average treatment effects with application to electronic health records data. *Biometrics*, 77(2):413–423, 2021.

B. De Finetti. La prévision: ses lois logiques, ses sources subjectives. In *Annales de l'institut Henri Poincaré*, volume 7, pages 1–68, 1937.

A. Dedieu, R. Mazumder, Z. Zhu, and H. Vahabi. Hierarchical modeling and shrinkage for user session lengthprediction in media streaming. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018.

T. Desautels, A. Krause, and J. W. Burdick. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *Journal of Machine Learning Research*, 2014.

W. Duan, S. Ba, and C. Zhang. Online experimentation with surrogate metrics: Guidelines and a case study. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 193–201, 2021.

E. Even-Dar, S. Mannor, Y. Mansour, and S. Mahadevan. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(6), 2006.

T. R. Fleming and J. H. Powers. Biomarkers and surrogate endpoints in clinical trials. *Statistics in medicine*, 31(25):2973–2984, 2012.

A. Grover, T. Markov, P. Attia, N. Jin, N. Perkins, B. Cheong, M. Chen, Z. Yang, S. Harris, W. Chueh, et al. Best arm identification in multi-armed bandits with delayed feedback. In *International conference on artificial intelligence and statistics*. PMLR, 2018.

C. N. Haddad. *Cholesky Factorization*, pages 374–377. Springer US, Boston, MA, 2009.

K. Han, S. Li, J. Mao, and H. Wu. Detecting interference in online controlled experiments with increasing allocation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 661–672, 2023.

D. Heath and W. Sudderth. De finetti's theorem on exchangeable variables. *The American Statistician*, 30(4):188–189, 1976.

B. Howson, C. Pike-Burke, and S. Filippi. Delayed feedback in generalised linear bandits revisited. In *International Conference on Artificial Intelligence and Statistics*, 2023.

G. W. Imbens and D. B. Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge university press, 2015.

L. Ji, Q. Qin, B. Han, and H. Yang. Reinforcement learning to optimize lifetime value in cold-start recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, 2021.

P. Joulani, A. Gyorgy, and C. Szepesvári. Online learning under delayed feedback. In *International conference on machine learning*, pages 1453–1461. PMLR, 2013.

N. Kallus and X. Mao. On the role of surrogates in the efficient estimation of treatment effects with limited outcome data. *arXiv preprint arXiv:2003.12408*, 2020.

Z. Karnin, T. Koren, and O. Somekh. Almost optimal exploration in multi-armed bandits. In *International conference on machine learning*, pages 1238–1246. PMLR, 2013.

R. Kohavi, D. Tang, and Y. Xu. *Trustworthy online controlled experiments: A practical guide to a/b testing*. Cambridge University Press, 2020.

M. Lalmas, H. O'Brien, and E. Yom-Tov. *Measuring user engagement*. Morgan & Claypool Publishers, 2014.

T. Lattimore and C. Szepesvári. An information-theoretic approach to minimax regret in partial monitoring. In *Conference on Learning Theory*, pages 2111–2139. PMLR, 2019.

T. Lattimore and C. Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

S. Li and S. Wager. Random graph asymptotics for treatment effect estimation under network interference. *The Annals of Statistics*, 50(4):2334 – 2358, 2022. doi: 10.1214/22-AOS2191.

Z. Liu, S. Liu, Z. Zhang, Q. Cai, X. Zhao, K. Zhao, L. Hu, P. Jiang, and K. Gai. Sequential recommendation for optimizing both immediate feedback and long-term retention. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, 2024.

T. Mandel, Y.-E. Liu, E. Brunskill, and Z. Popović. The queue method: Handling delay, heuristics, prior data, and evaluation in bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

L. Maystre, D. Russo, and Y. Zhao. Optimizing audio recommendations for the long-term: A reinforcement learning perspective. *arXiv preprint arXiv:2302.03561*, 2023.

T. M. McDonald, L. Maystre, M. Lalmas, D. Russo, and K. Ciosek. Impatient bandits: Optimizing recommendations for the long-term without delay. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, 2023.

C. D. Meyer, Jr. Generalized inverses of triangular matrices. *SIAM Journal on Applied Mathematics*, 18(2):401–406, 1970.

K. P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

S.-L. T. Normand. Meta-analysis: formulating, evaluating, combining, and reporting. *Statistics in medicine*, 18(3):321–359, 1999.

C. Pike-Burke, S. Agrawal, C. Szepesvari, and S. Grunewalder. Bandits with delayed, aggregated anonymous feedback. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4105–4113. PMLR, 2018.

R. L. Prentice. Surrogate endpoints in clinical trials: definition and operational criteria. *Statistics in medicine*, 8(4):431–440, 1989.

C. Qin and D. Russo. Adaptive experimentation in the presence of exogenous nonstationary variation, 2023.

C. E. Rasmussen, C. K. Williams, et al. *Gaussian processes for machine learning*. Springer, 2006.

L. Richardson, A. Zito, D. Greaves, and J. Soriano. Pareto optimal proxy metrics. 2023.

R. Rosenthal, H. Cooper, L. Hedges, et al. Parametric measures of effect size. *The handbook of research synthesis*, 621(2):231–244, 1994.

D. B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

D. Russo and B. Van Roy. Learning to optimize via information-directed sampling. *Advances in neural information processing systems*, 27, 2014.

D. Russo and B. Van Roy. An information-theoretic analysis of thompson sampling. *Journal of Machine Learning Research*, 17(68):1–30, 2016.

D. Russo, B. V. Roy, A. Kazerouni, I. Osband, and Z. Wen. A tutorial on thompson sampling, 2020.

S. Shahrampour, M. Noshad, and V. Tarokh. On sequential elimination algorithms for best-arm identification in multi-armed bandits. *IEEE Transactions on Signal Processing*, 2017.

M. Simchowitz, C. Tosh, A. Krishnamurthy, D. J. Hsu, T. Lykouris, M. Dudik, and R. E. Schapire. Bayesian decision-making under misspecified priors with applications to meta-learning. *Advances in Neural Information Processing Systems*, 34:26382–26394, 2021.

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

T. S. Thune, N. Cesa-Bianchi, and Y. Seldin. Nonstochastic multiarmed bandits with unrestricted delays. *Advances in Neural Information Processing Systems*, 32, 2019.

N. Tripuraneni, L. Richardson, A. D'Amour, J. Soriano, and S. Yadlowsky. Choosing a proxy metric from past experiments. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5803–5812, 2024.

T. J. VanderWeele. Surrogate measures and consistent surrogates. *Biometrics*, 2013.

C. Vernade, A. Carpentier, T. Lattimore, G. Zappella, B. Ermis, and M. Brueckner. Linear bandits with stochastic delayed feedback. In *International Conference on Machine Learning*, 2020.

Y. Wang, M. Sharma, C. Xu, S. Badam, Q. Sun, L. Richardson, L. Chung, E. H. Chi, and M. Chen. Surrogate for long-term user experience in recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

W. S. Weintraub, T. F. Lüscher, and S. Pocock. The perils of surrogate endpoints. *European heart journal*, 36(33):2212–2218, 2015.

H. Wu and S. Wager. Partial likelihood thompson sampling. In *Uncertainty in Artificial Intelligence*, pages 2138–2147. PMLR, 2022.

Q. Wu, H. Wang, L. Hong, and Y. Shi. Returning is believing: Optimizing long-term user engagement in recommender systems. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, 2017.

R. Xu, J. Bhandari, D. Korenkevych, F. Liu, Y. He, A. Nikulkov, and Z. Zhu. Optimizing long-term value for auction-based recommender systems via on-policy reinforcement learning. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys '23, 2023.

J. Yang, D. Eckles, P. Dhillon, and S. Aral. Targeting for long-term outcomes. *Management Science*, 70(6):3841–3855, 2024.

K. W. Zhang, H. Namkoong, D. Russo, et al. Posterior sampling via autoregressive generation. 2024.

V. Zhang, M. Zhao, A. Le, and N. Kallus. Evaluating the surrogate index as a decision-making tool using 200 a/b tests at netflix. *arXiv preprint arXiv:2311.11922*, 2023.

X. Zhang, H. Jia, H. Su, W. Wang, J. Xu, and J.-R. Wen. Counterfactual reward modification for streaming recommendation with delayed feedback. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, 2021.

G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li. Drn: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, 2018.

Z. Zhou, R. Xu, and J. Blanchet. Learning in generalized linear contextual bandits with stochastic delays. In *Advances in Neural Information Processing Systems*, 2019.

L. Zou, L. Xia, Z. Ding, J. Song, W. Liu, and D. Yin. Reinforcement learning to optimize long-term user engagement in recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.