Planning Contextual Adaptive Experiments with Model Predictive Control

Ethan Che ECHE25@GSB.COLUMBIA.EDU
Jimmy Wang JW4209@COLUMBIA.EDU
Hongseok Namkoong NAMKOONG@GSB.COLUMBIA.EDU
Decision, Risk and Operations
Columbia Business School
New York, NY, 10027 USA

Abstract

Implementing adaptive experimentation methods in the real world often encounters a multitude of operational difficulties, including batched/delayed feedback, non-stationary environments, and constraints on treatment allocations. To improve the flexibility of adaptive experimentation, we propose a Bayesian, optimization-based framework founded on model-predictive control (MPC) for the linear-contextual bandit setting. While we focus on simple regret minimization, the framework can flexibly incorporate multiple objectives along with constraints, batches, personalized and non-personalized policies, as well as predictions of future context arrivals. Most importantly, it maintains this flexibility while guaranteeing improvement over non-adaptive A/B testing across all time horizons, and empirically outperforms standard policies such as Thompson Sampling. Overall, this framework offers a way to guide adaptive designs across the varied demands of modern large-scale experiments.

Keywords: Adaptive Experimentation, Model Predictive Control, A/B Testing

1. Introduction

Experimentation is a core component of the deployment life-cycle of machine learning (ML) models in the real world, particularly for large-scale Internet platforms. These systems are often calibrated by a multitude of hyperparameters, ranging from model parameters to preferences over different metrics and objectives. These choices may influence system behavior in an unpredictable, black-box manner. This necessitates experimentation to carefully evaluate them and ensure quality of service.

Given the large number of possible configurations, adaptive experimentation offers a powerful methodology for efficiently allocating sampling effort. However, modern experiments increasingly involve many considerations and constraints, making it difficult to design adaptive methods in an appropriate manner. To name a few: first, experiments are almost always conducted in large batches with infrequent updates to the sampling policy (Bakshy et al., 2018; Offer-Westort et al., 2020; Jobjörnsson et al., 2022). Second, given the wide heterogeneity in how users respond to treatments, non-stationarity in the arrivals of users to the experiment can derail adaptive sampling policies (Qin and Russo, 2023). Finally, these experiments often must meet certain constraints, such as requiring sufficient sample coverage for post-experiment inference (Offer-Westort et al., 2020; Zhang et al., 2020), re-

quiring fairness (Chen et al., 2020), and ensuring safety (Amani et al., 2019). Given the difficulties of operationalizing adaptive algorithms, almost all experiments are non-adaptive A/B tests (Sculley et al., 2015; Agarwal et al., 2016).

In this work, we introduce a flexible optimization-based framework for designing adaptive experiments in a linear contextual bandit setting, in order to model personalization and non-stationarity. Under large batches, asymptotic normality of standard estimators allow the experimenter to update a tractable Bayesian model of their uncertainty. Using this model, we apply the model-predictive control (MPC) design principle to adaptively plan the sampling allocations: after observing new samples, re-solve for an optimal *non-adaptive* sampling policy to use for the rest of experiment. By resolving after every batch, the policy remains adaptive while working flexibly with batched feedback. Since the policy is the outcome of an optimization sub-routine, we can flexibly incorporate constraints. Finally, it allows the experimenter to incorporate predictions of context arrivals, ensuring robustness against predictible non-stationarity (e.g. day-of-the-week, seasonality).

Related Work Our work is most related to Che and Namkoong (2023), which models batched experiments as an MDP and proposes a model-predictive control policy for the multi-armed bandit. This work extends their approach to the linear contextual setting. In this regard, it is also closely related to dynamic programming approaches in Bayesian optimization (Frazier et al., 2008; Gonzalez et al., 2016; Lam et al., 2016; Wu and Frazier, 2019; Jiang et al., 2020). The considered objective of this work is simple/policy regret in linear contextual bandits, which has been studied by Ruan et al. (2021); Deshmukh et al. (2020); Zanette et al. (2021); Krishnamurthy et al. (2023). Our work builds on this literature by providing a Bayesian, optimization-based framework which can handle constraints in batch settings. In this sense, our work is also related to Jörke et al. (2022), which trains non-adaptive policies to minimize Bayesian simple regret, and Qin and Russo (2023), who study Bayesian bandit algorithms in the linear contextual setting which are robust to non-stationary variation. We propose a policy which also has a performance guarantee against uniform sampling, but which requires additional knowledge of context arrivals.

2. Model

We consider the linear contextual bandit model with K actions. Each experimental unit has a context $x \in \mathbb{R}^p$ and the reward $r(x, a)$ of assigning an action a to x is

$$r(x, a) = \phi(x, a)^\top \beta^* + \epsilon_i, \tag{1}$$

where $\phi(x_i, a_i) : \mathcal{X} \times [K] \to \mathbb{R}^d$ is a known feature mapping, $\beta^* \in \mathbb{R}^d$ is an unknown parameter vector and ϵ_i are iid mean zero $\mathbb{E}[\epsilon_i] = 0$ random variables with fixed, known measurement variance $\mathrm{Var}(\epsilon_i) = s^2$. This model can flexibly incorporate a variety of interactions between actions and contexts:

While our formulation allows for a variety of objectives, we are primarily interested in the best-policy identification task, in which an experiment is conducted across T rounds in order to identify an optimal policy π_T for assigning actions to users drawn from a reference distribution μ. If the experimenter knew β^*, the optimal policy would be

$$V_T^* := \max_{\pi_T} \mathbb{E}_{X\sim\mu}\left[\pi_T(a|X)\phi(X, a)^\top \beta^*\right] = \mathbb{E}_{X\sim\mu}\left[\max_a \phi(X, a)^\top \beta^*\right]$$

Given that the experimenter only observes noisy estimates of β^*, their objective is to obtain a policy $\hat{\pi}_T$ that minimizes **policy regret** (also known as **simple regret)** compared to the optimal policy.

$$\text{PolicyRegret}_T := V_T^* - \mathbb{E}_{X\sim\mu}\left[\hat{\pi}_T(a|X)\phi(X,a)^\top\beta^*\right]$$

Batched Experiment Design The experiment is composed of T sequential epochs (or "batches"). Within each epoch $t = 0, ..., T-1$,

1. A batch of n_t units arrives with contexts $\{X_i^t\}_{i=1}^{n_t}$ drawn iid from a distribution μ_t (which may change over time and differ from the reference distribution μ)
2. The experimenter selects a sampling allocation policy $\pi_t : \mathcal{X} \to \Delta_K$ based on previous observations and the contexts of the current batch.
3. Each unit is assigned to an action randomly according to $A_i \sim \pi_t(X_i^t)$.
4. The experimenter observes features $\Phi_t = \{\phi(X_i^t, A_i)\}_{i=1}^{n_t}$ and rewards $R_t = \{R_{t,i}\}_{i=1}^{n_t}$.

Given that the rewards follow a linear model, it is natural for the experimenter to estimate β^* through the ordinary least squares (OLS) estimator,

$$\hat{\beta}_t = (\Phi_t^\top\Phi_t)^{-1}\Phi_t R_t.$$

The signal-to-noise of this estimate will depend on the population design matrix Γ_t under sampling allocation π:

$$\Gamma_t(\pi) = \mathbb{E}_{X_i\sim\mu_t}\left[\sum_{a\in[K]}\pi(a|X_i)\phi(X_i,a)\phi(X_i,a)^\top\right]$$

When the batch size $n_t \to \infty$ is large, the OLS estimator is asymptotically normal by the central limit theorem. As long as $\Gamma_t(\pi)$ is invertible, then $\sqrt{n_t}(\hat{\beta}_t - \beta^*) \Rightarrow N(0, s^2\Gamma_t(\pi)^{-1})$. This justifies the normal approximation for the OLS estimator, which is the backbone of standard inferential procedures and power calculations:

$$\hat{\beta}_t \approx N\left(\beta^*, s^2(n_t\Gamma_t(\pi))^{-1}\right). \tag{2}$$

Bayesian Linear Model We consider an experimenter with a prior belief over β^*, $N(\beta_0, \Sigma_0)$ with mean $\beta_0 \in \mathbb{R}^d$ and $\Sigma_0 \in \mathbb{R}^{d\times d}$, possibly informed by a reservoir of previous experiments or domain knowledge. After each epoch t, the experimenter updates their posterior beliefs (β_t, Σ_t) after assigning actions according to π_t and calculating the OLS estimator $\hat{\beta}_t$. The experimenter updates their posterior using the normal approximation (2), updating $\Sigma_{t+1} = \left(s^{-2}n_t\Gamma_t(\pi_t) + \Sigma_t^{-1}\right)^{-1}$ and $\beta_{t+1} = \Sigma_{t+1}(\Sigma_t^{-1}\beta_t + s^{-2}n_t\Gamma_t(\pi_t)\hat{\beta}_t)$.

We can think of the belief (β_t, Σ_t) as the states of a Markov Decision Process (MDP), where states transitions are determined by the sampling allocation π_t. We can reparameterize the state transition using the posterior predictive distribution of $\hat{\beta}_t$ given β_t, Σ_t.

Proposition 1 *Let $Z_1, ..., Z_T$ be standard iid $N(0, I_d)$ random vectors. The system governed by the posterior updates has the same joint distribution as that with the following state transition:*

$$\Sigma_{t+1} = \left(s^{-2}n_t\Gamma_t(\pi_t) + \Sigma_t^{-1}\right)^{-1} \tag{3}$$

$$\beta_{t+1} = \beta_t + (\Sigma_t - \Sigma_{t+1})^{1/2} Z_t \tag{4}$$

The proof is in Appendix A.1. The state transitions (3) quantifies the future reduction of uncertainty given a sampling allocation π_t, and can be rolled out for several steps. Crucially, this framework offers a *smoothed* model of partial feedback, for which state transitions are differentiable in the sampling allocation π_t (as $\Gamma_t(\pi)$ is linear in π). This enables the use of autodifferentation frameworks (e.g. PyTorch (Paszke et al., 2019) or Tensorflow (Abadi et al., 2016)) to calculate gradients of objectives with respect to the sampling allocation along a sample path $(Z_0, ..., Z_T)$, and optimize via gradient descent.

Bayesian Objective The experimenter's objective consists of a sum of per-period rewards v_s, which can depend on the posterior state (β_s, Σ_s) and the sampling allocation π_s at period s. Given a sampling policy $\pi = \{\pi_t(\cdot|\beta_t, \Sigma_t)\}_{t=0}^T$, the value function at time t is

$$V_t^\pi(\beta_t, \Sigma_t) = \mathbb{E}\left[\sum_{s=t}^{T} v_s(\pi_s, \beta_s, \Sigma_s)\right].$$

If the experimenter is maximizing policy reward or minimizing policy regret, then they would only experience a terminal reward v_T (i.e. $v_s = 0$ for $s < T$). One can show that minimizing Bayesian policy regret is equivalent to maximizing the policy reward under the policy that acts greedily according to final posterior (β_T, Σ_T):

$$v_T(\pi_T, \beta_T, \Sigma_T) = \mathbb{E}_{X\sim\mu}\left[\max_a \phi(X, a)^\top \beta_T\right]$$

3. Residual Horizon Optimization

Using the model (3), we can solve for an optimal policy using dynamic programming. However, this can be computationally expensive, considering the high-dimensional state and action spaces. We describe a computationally feasible method denoted as Residual Horizon Optimization (RHO), introduced in Che and Namkoong (2023). The policy is based on model-predictive control (MPC): at every epoch t, solve for an optimal *static* trajectory $\{\rho_t, ..., \rho_T\}$ of sampling allocations given the current state (β_t, Σ_t). This policy has many advantages:

- Efficient to solve with stochastic gradient descent (SGD) and function approximation.
- Flexibility with different objectives and constraints.
- The planning problem calibrates exploration with the length of the remaining horizon.
- The experimenter can use knowledge of future context distributions $\mu_t, ..., \mu_T$ to ensure robustness to non-stationarity (Proposition 2).

Residual Horizon Optimization Concretely, the RHO policy $\rho_t^*(\beta_t, \Sigma_t)$ selects ρ_t^* from a sequence of sampling policies $\rho_t^*, ..., \rho_T^*$ that maximizes the following objective

$$\begin{aligned} &\underset{\rho_t,...,\rho_T}{\text{maximize}} \;\; V_t^{\rho_{t:T}}(\beta_t, \Sigma_t) = \mathbb{E}_t\left[\sum_{s=1}^{T} v_s(\rho_s, \beta_s, \Sigma_s)\right] \\ &\text{subject to} \;\; g_t(\rho_s) \le c_t, \quad \forall s = t, ..., T \end{aligned} \tag{5}$$

where g_t are convex constraint functions and $c_t \in \mathbb{R}^h$. When the experimenter's objective is policy regret, we can show that it is sufficient to optimize over a single allocation ρ_t (see Appendix A.2).

$$\begin{aligned} &\underset{\rho_t}{\text{maximize}} \ \mathbb{E}_{t,X\sim\mu}\left[\max_a \phi(X,a)^\top \beta_t + \sqrt{\phi(X,a)^\top \left[\Sigma_t - \Sigma_T(\rho_t)\right]\phi(X,a)}Z\right] \\ &\text{subject to } \ g_t(\rho_t) \le c_t \end{aligned} \tag{6}$$

where $\Sigma_T(\rho_t) = (\Sigma_t^{-1} + \sum_{s=t}^{T-1} n_s \Gamma_s(\rho_t))^{-1}$ is the posterior covariance after sampling with ρ_t across epochs $t, .., T$. Importantly, the overall objective is differentiable in ρ_t.

Evaluating the objective requires knowledge of the measurement variance s^2 and the sum of population covariances $\sum_{s=t}^{T-1} n_s \Gamma_s(\rho_s)$, which in turn requires knowing the context distributions μ_t. In practice, to solve the above problem one would

1. Sample $Z \sim N(0,1)$ and contexts from μ to approximate expectation in (6).
2. Parameterize the sampling allocation $\rho_t^\theta(a|x)$, e.g. multi-layer perceptron.
3. Use current batch $\{X_i^t\}_{i=1}^{n_t}$ and samples from future context distributions $\mu_s, s \ge t$ to estimate population covariances in $\Sigma_T(\rho_t^\theta)$.

$$n_s \Gamma_s(\rho_t^\theta) \approx \sum_{i=1}^{n_s}\sum_{a=1}^{K} \rho_t^\theta(a|X_i^s)\phi(X_i^s,a)\phi(X_i^s,a)^\top, \quad X_i^s \sim \mu_s, \quad \forall s = t, ..., T-1$$

4. Optimize (6) by stochastic gradient descent over policy parameters θ.

The experimenter can use offline data (as in Zanette et al. (2021)) to estimate future context arrivals, especially if the non-stationarity of contexts can be planned for (e.g. day of the week). The structure of this policy immediately gives guarantees on the performance, which motivates the MPC design principle. The proof is in Appendix A.3.

Proposition 2 *(Policy Improvement) Consider any static sequence $\bar{\rho} = (\bar{\rho}_0, ..., \bar{\rho}_T)$ of sampling allocation policies, which are dynamically feasible: $g_t(\bar{\rho}_s) \le c_t, \forall s \ge t \in [T]$. Let $V_t^{\bar{\rho}_{0:T}}$ be the corresponding value function under the Bayesian model and $V_t^{\rho^*}(\beta_t, \Sigma_t)$ be the value function of the RHO policy that solves* (5)*. We have that for all t, β_t, Σ_t that*

$$V_t^{\rho^*}(\beta_t, \Sigma_t) \ge \max_{\bar{\rho}} V_t^{\bar{\rho}}(\beta_t, \Sigma_t)$$

This is a robustness property that guarantees performance improvement over A/B testing for *all* epochs t, even under non-stationary contexts. This holds even with constraints. In contrast, adapting standard heuristics (e.g. Thompson Sampling) to constraints requires ad hoc adjustments and new, bespoke proofs of convergence. These constraints typically concern the sampling allocation for units in the current batch $\{X_i^t\}_{i=1}^{n_t}$. Some examples include:

$$g_{\text{budget}}(\rho_t) = \sum_{i=1}^{n_t} c^\top \rho_t(X_i^t) \le B, \quad g_{\text{safe}}(\rho_t) = \log \mathbb{P}_t\left(-\sum_{i=1}^{n_t} \rho_t(X_i^t)^\top r(X_i^t) \le \bar{r}\right) \ge \log 0.99$$

where $\rho_t(X_i^t) \in \Delta_K$ and $r(X_i^t) \in \mathbb{R}^K$ are the sampling probabilities and possible rewards for individual X_i^t. The constraint g_{budget} is a budget constraint with action costs $c \in \mathbb{R}^K$, and g_{safe} is a safety constraint that places an lower bound on the probability that the total reward is above a value $-\bar{r} > 0$ (under a Gaussian approximation, this is concave in ρ_t).

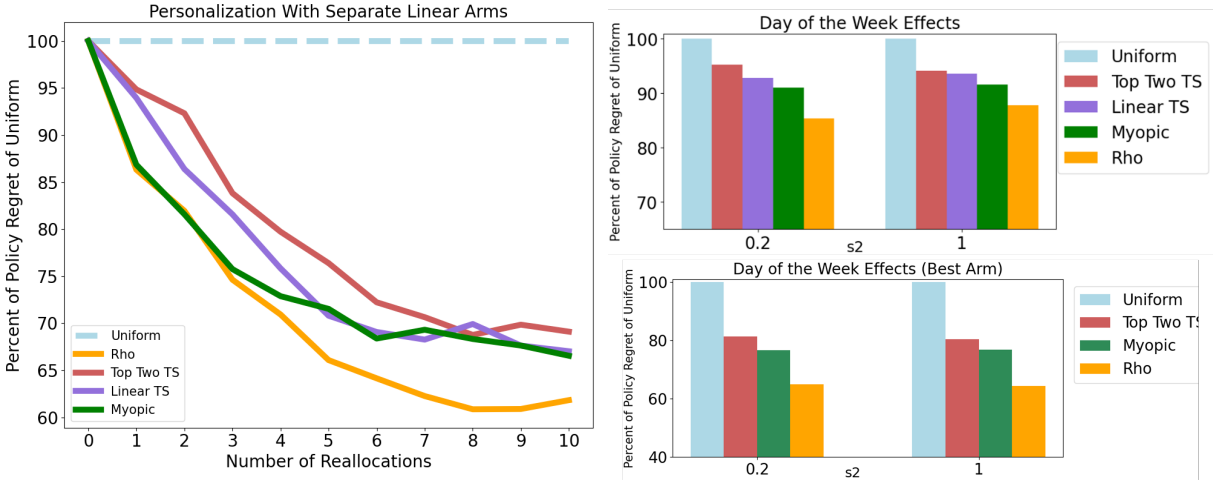

Figure 1: (Left) Relative gains over uniform sampling in personalization example across a range of time horizons T (batch size = 100, Gumbel noise with $s^2 = 0.2$). (Right) Relative gains over uniform sampling in day-of-the-week example for $s^2 \in \{0.2, 1\}$. Above figure shows policy regret when the final policy assigns an arm for each day. Below shows simple regret for a final policy that selects a single action.

4. Experiments

Personalization We evaluate the policy regret of different experimentation methods for a setting in which the optimal policy gives a personalized action for each user. We simulate an environment with 10 arms, each with an unknown parameter β_a and rewards given as,

$$r(x_i, a) = x_i^\top \beta_a + \epsilon_i. \tag{7}$$

We compare against Thompson Sampling (Agrawal and Goyal, 2013), a Myopic version of RHO, and Top-Two Thompson Sampling (Qin and Russo, 2023). In Figure 4, RHO exhibits large gains over other benchmarks, which holds under different noise levels and even heavy-tailed (Student-T) and skewed (Gumbel) noise ϵ_i. Details can be found in B.

Robustness to Nonstationarity Standard adaptive algorithms have been shown to fail under nonstationarity (Qin and Russo, 2023). Similar to (Qin and Russo, 2023), we consider a day of the week effect model in which there are $|\mathcal{X}| = 7$ contexts and

$$r(x, a) = \theta_x + \theta_{x,a} + \theta_a + \epsilon_i \tag{8}$$

This setting consists of one batch experiment per day for seven days with day-specific rewards. There are 10 actions and we consider two objectives: choosing an action to deploy for all days (best-arm identification), and choosing an action to deploy for each specific day (best-policy identification). We observe that major advantage of the RHO policy is that it can plan for future non-stationarity, as it uses the sum of (known) future population covariances $\sum_{s=t}^{T-1} n_s \Gamma_s(\rho_s)$, which is feasible since the days in the week are known in advance. Figure 1 shows that RHO outperforms in both objectives, and additional details can be found in Section B.

Appendix A. Proofs

A.1 Proof of Proposition 1

We can simplify the Bayesian posterior update for the mean as follows:

$$\begin{aligned}&\Sigma_{t+1}(\Sigma_t^{-1}\beta_t + s^{-2}n_t\Gamma_t(\pi)\beta_t - s^{-2}n_t\Gamma_t(\pi)\beta_t + s^{-2}n_t\Gamma_t(\pi)\hat{\beta}_t)\\ &= \beta_t + \Sigma_{t+1}(s^{-2}n_t\Gamma_t(\pi))(\hat{\beta}_t - \beta_t)\end{aligned}$$

Note that the posterior predictive distribution of $\hat{\beta}_t \sim N(\beta^*, s^2(n_t\Gamma_t(\pi))^{-1})$ is given by

$$\begin{aligned}\hat{\beta}_t &= \beta^* + s(n_t\Gamma_t)^{-1/2}Z_1\\ &= (\beta_t + \Sigma_t^{1/2}Z_2) + s(n_t\Gamma_t)^{-1/2}Z_1\end{aligned}$$

where Z_1 and Z_2 are independent $N(0, I_d)$ random vectors. This implies

$$\begin{aligned}\text{Var}\left((s^{-2}n_t\Gamma_t)(\hat{\beta}_t - \beta_t)\right) &= \text{Var}\left((s^{-2}n_t\Gamma_t)(\Sigma_t^{1/2}Z_2 + s(n_t\Gamma_t)^{-1/2}Z_1)\right)\\ &= s^{-4}n_t^2\left(\Gamma_t\Sigma_t\Gamma_t\right) + \left(s^{-2}n_t\Gamma_t\right)\left(s^2n_t^{-1}\Gamma_t^{-1}\right)\left(s^{-2}n_t\Gamma_t\right)\\ &= s^{-4}n_t^2\left(\Gamma_t\Sigma_t\Gamma_t\right) + \left(s^{-2}n_t\Gamma_t\right)\end{aligned}$$

To compute $\text{Var}\left(\Sigma_{t+1}(s^{-2}n_t\Gamma_t)(\hat{\beta}_t - \beta_t)\right)$, we observe that

$$\begin{aligned}\text{Var}\left(\Sigma_{t+1}(s^{-2}n_t\Gamma_t)(\hat{\beta}_t - \beta_t)\right) &= \Sigma_{t+1}\left(s^{-4}\left((n_t\Gamma_t)\Sigma_t(n_t\Gamma_t)\right) + \left(s^{-2}n_t\Gamma_t\right)\right)\Sigma_{t+1}\\ &= \Sigma_{t+1}\left(s^{-2}n_t\Gamma_t\Sigma_t + I\right)\left(s^{-2}n_t\Gamma_t\right)\Sigma_{t+1}\end{aligned}$$

We use the identity $(A+B)^{-1} = A^{-1} - (A + AB^{-1}A)^{-1}$, taking $A = s^{-2}n_t\Gamma_t$ and $B = \Sigma_t^{-1}$ and observe that

$$\begin{aligned}\Sigma_{t+1} &= (s^{-2}n_t\Gamma_t)^{-1} - \left[s^{-4}\left((n_t\Gamma_t)\Sigma_t(n_t\Gamma_t)\right) + \left(s^{-2}n_t\Gamma_t\right)\right]^{-1}\\ &= A^{-1} - \left[(AB^{-1} + I)A\right]^{-1}\end{aligned}$$

Further simplifying, we have

$$\begin{aligned}&\Sigma_{t+1}\left(s^{-4}\left((n_t\Gamma_t)\Sigma_t(n_t\Gamma_t)\right) + \left(s^{-2}n_t\Gamma_t\right)\right)\Sigma_{t+1}\\ &= \left(A^{-1} - \left[(AB^{-1} + I)A\right]^{-1}\right)\left((AB^{-1} + I)A\right)\left(A^{-1} - \left[(AB^{-1} + I)A\right]^{-1}\right)\\ &= \left(A^{-1} - \left[(AB^{-1} + I)A\right]^{-1}\right)\left((AB^{-1} + I) - I\right)\\ &= \left(A^{-1} - \left[(AB^{-1} + I)A\right]^{-1}\right)AB^{-1}\\ &= \left(A^{-1} - \left[(AB^{-1} + I)A\right]^{-1}\right)AB^{-1}\\ &= \Sigma_{t+1}s^{-2}n_t\Gamma_t\Sigma_t\end{aligned}$$

Finally, we can observe that $\Sigma_{t+1}s^{-2}n_t\Gamma_t\Sigma_t$ is equal to $\Sigma_t - \Sigma_{t+1}$. Since, $\Sigma_{t+1}(s^{-2}n_t\Gamma_t)(\hat{\beta}_t - \beta_t)$ is a mean-zero Gaussian random vector, it can be expressed as $\text{Var}\left(\Sigma_{t+1}(s^{-2}n_t\Gamma_t)(\hat{\beta}_t - \beta_t)\right)^{1/2} Z$. So altogether the posterior update can be expressed as

$$(\Sigma_t - \Sigma_{t+1})^{1/2} Z_t$$

A.2 Derivation of (6)

Conditional on the current posterior (β_t, Σ_t), the final posterior mean after sampling with static policies $\rho_t, ..., \rho_{T-1}$ is

$$\beta_t + \sum_{s=t}^{T-1} (\Sigma_s - \Sigma_{s+1})^{1/2} Z_s$$

where $\Sigma_{s+1} = (\Sigma_t^{-1} + \sum_{u=t}^{s} n_u \Gamma_u(\rho_u))^{-1}$. Since Z_s are independent and ρ_s are non-adaptive, this is a sum of independent random vectors and by the reparameterization trick:

$$\begin{aligned}\sum_{s=t}^{T-1} (\Sigma_s - \Sigma_{s+1})^{1/2} Z_s &\overset{d}{=} \left(\sum_{s=t}^{T-1} \Sigma_s - \Sigma_{s+1}\right)^{1/2} Z \\ &= (\Sigma_t - \Sigma_T)^{1/2} Z\end{aligned}$$

We can write the objective then as

$$\mathbb{E}_t\left[\max_a \phi(x,a)^\top \beta_T\right] = \mathbb{E}_t\left[\max_a \phi(x,a)^\top \beta_t + \sqrt{\phi(x,a)^\top (\Sigma_t - \Sigma_T)\phi(x,a)} Z\right]$$

Note that $\Gamma_u(\rho_u)$ is linear in ρ_u for all u, Σ_T depends on the sampling allocations only through the sum. This means that by replacing $\rho_t, ..., \rho_{T-1}$ with a the same allocation policy at each round $\tilde{\rho}_t, ..., \tilde{\rho}_t$ where $\tilde{\rho}_t = \frac{1}{T-t-1}\sum_{s=t}^{T-1} \rho_t$ is the average of $\rho_t, ..., \rho_{T-1}$, one can achieve the exact same objective value. This will also be feasible since $\tilde{\rho}_t$ will still map to Δ_K and since $g_t(\rho_s) \leq c_t$ for all ρ_s then by Jensen's inequality, $g_t(\tilde{\rho}_t)$ will also be less than c_t.

A.3 Proof of Proposition 2

First observe that when at T and $T-1$, the static policy and RHO coincide so $V_t^{\rho^*}(\beta_t, \Sigma_t) = \max_{\bar{\rho}} V_t^{\bar{\rho}}(\beta_t, \Sigma_t)$. Next, as an induction hypothesis, suppose for all $(\beta_{t+1}, \Sigma_{t+1})$,

$$V_{t+1}^{\rho^*}(\beta_{t+1}, \Sigma_{t+1}) \geq \max_{\bar{\rho}_{t+1:T}} V_{t+1}^{\bar{\rho}_{t+1:T}}(\beta_{t+1}, \Sigma_{t+1}).$$

Then for any (β_t, Σ_t),

Policy	$s2 = 0.2, \sigma^2 = 0.1$	$s2 = 0.2, \sigma^2 = 5$	$s2 = 1, \sigma^2 = 0.1$	$s2 = 1, \sigma^2 = 5$
(Gumbel) Linear TS	67.0	82.8	94.22	91.7
Linear Top-Two TS	69.1	93.8	93.8	94.0
Myopic	66.5	**79.9**	83.1	86.5
RHO	**61.8**	80.3	**81.1**	**86.1**
(Student's t) Linear TS	77.1	102	90.6	91.6
Linear Top-Two TS	76.1	97.8	90.8	88.9
Myopic	64.5	83.1	70.0	85.77
RHO	**60.5**	**83.0**	**66.5**	**83.3**

Figure 2: Percent of policy regret of uniform in the personalization setting under multiple combinations of context variance {0.1, 5}, measurement variance {0.2, 1}, and noise distribution {Gumbel, Student's t}

$$
\begin{aligned}
V_t^{\rho^*}(\beta_t, \Sigma_t) &= v_t(\rho_t^*, \beta_t, \Sigma_t) + \mathbb{E}_t[V_{t+1}^{\rho^*}(\beta_{t+1}, \Sigma_{t+1})] \\
&\geq v_t(\rho_t^*, \beta_t, \Sigma_t) + \mathbb{E}_t\left[\max_{\bar{\rho}_{t+1:T}} V_{t+1}^{\bar{\rho}_{t+1:T}}(\beta_{t+1}, \Sigma_{t+1})\right] \\
&\geq v_t(\rho_t^*, \beta_t, \Sigma_t) + \max_{\bar{\rho}_{t+1:T}} \mathbb{E}_t[V_{t+1}^{\bar{\rho}_{t+1:T}}(\beta_{t+1}, \Sigma_{t+1})] \\
&= \max_{\bar{\rho}_{t:T}} \left\{ v_t(\bar{\rho}_t, \beta_t, \Sigma_t) + \mathbb{E}_t[V_{t+1}^{\bar{\rho}_{t+1:T}}(\beta_{t+1}, \Sigma_{t+1})] \right\} \\
&= \max_{\bar{\rho}_{t:T}} V_t^{\bar{\rho}_{t:T}}(\beta_t, \Sigma_t)
\end{aligned}
$$

using the definition for the policy ρ_t^*.

Appendix B. Experimental Details

Personalization To incorporate the objective 7 into the flexible reward model 1, the parameter vector $\beta^* = (\beta_a^*)_{a=1}^K \in \mathbb{R}^{Kp}$ specifies an embedding vector $\beta_a^* \in \mathbb{R}^p$ for each action where the reward is given by the dot product,

$$\phi(x, a)^\top \beta^* = x^\top \beta_a^*. \tag{9}$$

In this simulation, there are 10 reallocation epochs and 10 possible arms to deploy, each corresponding to a separate β_a^*. We let the context be of dimension 5 and each β^* is normally distributed with diagonal variance 0.001. The contexts are also distributed normally according to $N(1, \sigma^2 I)$, where we vary the level of σ^2 from 0.1 to 5. We choose to center the distribution at mean 1 because the optimal sampling policy for a mean 0 context distribution is to sample each arm equally. In our setup, a greater context variance leads to a higher support of optimal arms, while small context variance converges to a treatment effect example. To test the robustness of the different methods, we also compare the method under different noise levels and distributions in 2. Specifically, we choose the Gumbel distribution and student's t distribution to test robustness under heavy-tailed and

skewed distributions. The high performance of RHO in each setting shows the validity of the OLS batch approximation under non-Gaussian noise distributions. Additionally, we can see that thompson sampling performs especially poor with high measurement variance or context variance

Day of the Week Effects The reward model 1 can also be adapted for 8, where the parameter vector $\beta^* = (\beta_x,\ \beta_{x,a},\ \beta_a) \in \mathbb{R}^{K+Kd+d}$ is composed of an action-specific vector $\beta_a \in \mathbb{R}^K$, a day-specific reward $\beta_x \in \mathbb{R}$, and a reward unique to the action and day $\beta_{x,a} \in \mathbb{R}^{Kd}$. Encoding the day and action into one-hot basis vectors $e_x \in \mathbb{R}^d$, $e_a \in \mathbb{R}^K$, $e_{x,a} \in \mathbb{R}^{Kd}$, the reward is given by the sum,

$$\phi(x,a)^\top \beta^* = \langle \beta_x, e_x \rangle + \langle \beta_{x,a}, e_{x,a} \rangle + \langle \beta_a, e_a \rangle. \tag{10}$$

The experimental setting consists of a batch of 100 samples arriving on each day of the week for seven days. β^* is distributed according to $N(0, \sigma^2 I)$ where $\sigma^2 = 0.001$, and in figure 1 we consider Gaussian noise with measurement variances 0.2 and 0.1. There are two natural objectives that an experimenter may utilize:

1. **Best Arm Identification**: The goal is to choose the single best arm to deploy across all days. This corresponds to minimizing the simple regret of an action a. The objective can then be formulated as

$$\max_{a \in \mathcal{A}} \{ \sum_{i=1}^{7} w(x_i) \phi(x_i, a)^\top \beta^* \}$$

   Let $\overline{\phi(a)} = \sum_{i=1}^{N} w(x_i)\phi(x_i, a)$. Since we are now optimizing for the mean of an action effect, we can adjust the RHO objective 6 to reflect this:

$$\mathbb{E}\left[\max_{a \in \mathcal{A}} \{ \overline{\phi(a)}^\top \beta_t + \sqrt{\overline{\phi(a)}^\top [\Sigma_t - \Sigma_T(\rho_t)] \overline{\phi(a)}} Z \} \right]$$

   We benchmark RHO against context unaware thompson sampling and the recently proposed "Deconfounded Thompson Sampling" (Qin and Russo, 2023), which optimizes the mean while updating its posterior using contexts. Figure 1 shows the superior performance of RHO compared to existing methods. It is worth noting that the current thompson sampling methods fail to exploit knowledge of future contexts.

2. **Best Arm For Each Day**: We also consider a setting where an experimeter can deploy different arms each day and wants to maximize the expected reward of doing so. The objective corresponds to 6, but this setup is non-trivial as the distribution of contexts (days) is non-stationary and deterministic. By planning for future context distributions, Figure 1 shows how RHO outperforms other benchmarks like linear top two thompson sampling. Empirically, we find that if RHO naively treats the context distribution as stationary, in that it plans using only the given contexts at one day, its performance is worse, showing the necessity of planning.

References

Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016. URL `https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf`.

Alekh Agarwal, Sarah Bird, Markus Cozowicz, Luong Hoang, John Langford, Stephen Lee, Jiaji Li, Dan Melamed, Gal Oshri, Oswaldo Ribas, et al. Making contextual decisions with low technical debt. *arXiv preprint arXiv:1606.03966*, 2016.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 127–135, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL `https://proceedings.mlr.press/v28/agrawal13.html`.

Sanae Amani, Mahnoosh Alizadeh, and Christos Thrampoulidis. Linear stochastic bandits under safety constraints. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 19*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper_files/paper/2019/file/09a8a8976abcdfdee15128b4cc02f33a-Paper.pdf`.

Eytan Bakshy, Lili Dworkin, Brian Karrer, Konstantin Kashin, Benjamin Letham, Ashwin Murthy, and Shaun Singh. Ae: A domain-agnostic platform for adaptive experimentation. In *Neural Information Processing Systems Workshop on Systems for Machine Learning*, pages 1–8, 2018.

Ethan Che and Hongseok Namkoong. Adaptive experimentation at scale: A computational framework for flexible batches. *arXiv:2303.11582 [cs.LG]*, 2023.

Yifang Chen, Alex Cuellar, Haipeng Luo, Jignesh Modi, Heramb Nemlekar, and Stefanos Nikolaidis. Fair contextual multi-armed bandits: Theory and experiments. In Jonas Peters and David Sontag, editors, *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 181–190. PMLR, 03–06 Aug 2020.

Aniket Anand Deshmukh, Srinagesh Sharma, James W. Cutler, Mark Moldwin, and Clayton Scott. Simple regret minimization for contextual bandits. *arXiv:1810.07371 [stat.ML]*, 2020.

Peter I. Frazier, Warren B. Powell, and Savas Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5): 2410–2439, 2008. doi: 10.1137/070693424.

Javier Gonzalez, Michael Osborne, and Neil Lawrence. Glasses: Relieving the myopia of bayesian optimisation. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 2016.

Shali Jiang, Daniel Jiang, Maximilian Balandat, Brian Karrer, Jacob Gardner, and Roman Garnett. Efficient nonmyopic bayesian optimization via one-shot multi-step trees. In *Advances in Neural Information Processing Systems 20*, 2020.

Sebastian Jobjörnsson, Henning Schaak, Oliver Musshoff, and Tim Friede. Improving the statistical power of economic experiments using adaptive designs. *Experimental Economics*, 2022.

Matthew Jörke, Jonathan Lee, and Emma Brunskill. Simple regret minimization for contextual bandits using bayesian optimal experimental design. In *ICML2022 Workshop on Adaptive Experimental Design and Active Learning in the Real World*, 2022.

Sanath Kumar Krishnamurthy, Ruohan Zhan, Susan Athey, and Emma Brunskill. Proportional response: Contextual bandits for simple and cumulative regret minimization. *arXiv:2307.02108 [cs.LG]*, 2023.

Remi R. Lam, Karen E. Willcox, and David H. Wolpert. Bayesian optimization with a finite budget: an approximate dynamic programming approach. In *Advances in Neural Information Processing Systems 16*, 2016.

Molly Offer-Westort, Alexander Coppock, and Donald P. Green. Adaptive experimental design: Prospects and applications in political science. *SSRN 3364402*, 2020. URL `http://dx.doi.org/10.2139/ssrn.3364402`.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, 2019.

Chao Qin and Daniel Russo. Adaptive experimentation in the presence of exogenous nonstationary variation. *arXiv:2202.09036 [cs.LG]*, 2023.

Yufei Ruan, Jiaqi Yang, and Yuan Zhou. Linear bandits with limited adaptivity and learning distributional optimal design. *arXiv:2007.01980 [cs.LG]*, 2021.

David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems 28*, pages 2503–2511, 2015.

Jian Wu and Peter I. Frazier. Practical two-step look-ahead bayesian optimization. In *Advances in Neural Information Processing Systems 19*, 2019.

Andrea Zanette, Kefan Dong, Jonathan N Lee, and Emma Brunskill. Design of experiments for stochastic contextual linear bandits. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 21*, volume 34, pages 22720–22731. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/c00193e70e8e27e70601b26161b4ae86-Paper.pdf.

Kelly Zhang, Lucas Janson, and Susan Murphy. Inference for batched bandits. *Advances in Neural Information Processing Systems 20*, 33:9818–9829, 2020.