

Multitask Learning and Bandits via Robust Statistics

Kan Xu

University of Pennsylvania, Economics, kanxu@sas.upenn.edu

Hamsa Bastani

Wharton School, Operations Information and Decisions, hamsab@wharton.upenn.edu

Decision-makers often simultaneously face many related but heterogeneous learning problems. For instance, a large retailer may wish to learn product demand at different stores to solve pricing or inventory problems, making it desirable to learn jointly for stores serving similar customers; alternatively, a hospital network may wish to learn patient risk at different providers to allocate personalized interventions, making it desirable to learn jointly for hospitals serving similar patient populations. Motivated by real datasets, we study a natural setting where the unknown parameter in each learning instance can be decomposed into a shared global parameter plus a sparse instance-specific term. We propose a novel two-stage multitask learning estimator that exploits this structure in a sample-efficient way, using a unique combination of *robust statistics* (to learn across similar instances) and *LASSO regression* (to debias the results). Our estimator yields improved sample complexity bounds in the feature dimension d relative to commonly-employed estimators; this improvement is exponential for “data-poor” instances, which benefit the most from multitask learning. We illustrate the utility of these results for online learning by embedding our multitask estimator within simultaneous contextual bandit algorithms. We specify a dynamic calibration of our estimator to appropriately balance the bias-variance tradeoff over time, improving the resulting regret bounds in the context dimension d . Finally, we illustrate the value of our approach on synthetic and real datasets.

Key words: multitask learning, transfer learning, robust statistics, sparsity, LASSO, contextual bandits

1. Introduction

Predictive analytics powers data-driven decision-making across many domains. However, many problems in practice suffer from “small data” – i.e., only a very limited quantity of labeled data is available from the target predictive task, hindering training of highly accurate predictive models. As a consequence, a common solution is to leverage training data from related (but different) predictive tasks to reduce variance. In other words, we have an opportunity to not only learn *within* each predictive task, but also *across* similar tasks. To illustrate, consider the following two examples from healthcare and revenue management respectively:

EXAMPLE 1 (MEDICAL RISK SCORING). Health providers seek to predict patient-specific risk for adverse events (e.g., diabetes) in order to target preventative interventions. To this end, in our experiments in §6, we use electronic medical record data to predict the likelihood of an upcoming

Type II diabetes diagnosis for patients. Learning this risk score primarily from patient data collected at the *target* hospital (where the patient is being seen and treatment decisions will be made) is important to account for idiosyncrasies that are specific to the hospital and the patient population it serves. Indeed, we find that a predictive model trained using electronic medical record data from one hospital performs quite poorly when evaluated on patients from other hospitals (see Figure 1 below), with the out-of-sample AUROC degrading from 0.8 at the target hospital to 0.5-0.65 at other hospitals. This is due to a well-known phenomenon called *dataset shift* (Quiñonero-Candela et al. 2008); in the medical context, this can arise due to systematic differences across hospitals in diagnosis/treatment behavior, healthcare utilization, or medical coding (see, e.g., Subbaswamy and Saria 2020, Bastani 2021, Mullainathan and Obermeyer 2017). Therefore, to obtain good performance for all patients, *each* hospital faces a distinct learning problem. Yet, we may expect hospitals that serve similar patient populations to have similar underlying predictive models, creating an important opportunity to transfer knowledge across problem instances.

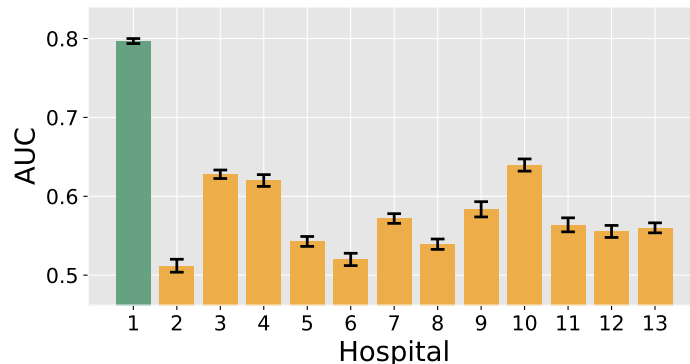


Figure 1 Out-of-sample performance (measured by AUROC) of a predictive model trained on electronic medical record data from Hospital #1 evaluated on patients from Hospital #1 (green) and Hospitals #2–13 (yellow); point estimates and 95% confidence intervals are based on 1000 random draws of the data. We observe a significant degradation in predictive performance in non-target hospitals due to dataset shift.

EXAMPLE 2 (DEMAND PREDICTION). Large retailers need to predict store-specific demand for their various products to inform dynamic pricing or inventory management decisions. Again, to protect from dataset shift, it is important to learn this demand model primarily from sales data collected at the *target* store (where sales occurred and decisions will be taken); this will account for idiosyncrasies that are specific to the store and the customer population it serves, including systematic differences in customer trends/preferences, in-store product placement, or promotion decisions (see, e.g., Baardman et al. 2020, Cohen and Perakis 2018, van Herpen et al. 2012). As a result, *each* store faces a distinct learning problem. Yet, we may expect stores that serve similar customer populations to have similar underlying demand models, creating an important opportunity to transfer knowledge across problem instances.

There are numerous other examples where we wish to learn predictive models across related tasks to inform targeted decision-making policies — e.g., customer promotion targeting for many different promotions, A/B testing on platforms for many candidate interventions, and clinical trials for many promising combination therapies. While the data-driven decision-making literature typically considers a single decision-maker solving an isolated problem instance, we focus on developing algorithms for the setting with many (potentially simultaneous) related learning tasks. We also extend our approach to online learning via simultaneous contextual bandits — predictive algorithms that are effective with “small data” are especially useful here because bandits are largely used in problems where there is little historical data available, e.g., due to the novelty or nonstationarity of the learning problem, or the limited population size relative to the feature dimension.

We build on the transfer and multitask learning literature (Caruana 1997, Pan et al. 2010), which proposes general algorithms to transfer knowledge across problem instances to improve learning. Unfortunately, these algorithms typically do not improve parameter recovery bounds (ignoring constants) — i.e., they do not significantly improve predictive performance compared to treating each learning task as its own independent problem. Indeed, in general, transfer or multitask learning cannot improve predictive accuracy without assuming some form of shared structure connecting the different problem instances — intuitively, if the predictive tasks are unrelated, then learning in one task cannot significantly improve learning in others (Hanneke and Kpotufe 2020). Our work bridges this gap by imposing a natural shared structure — *sparse heterogeneity* — motivated via real datasets. By designing a multitask estimator that efficiently exploits this structure, we obtain improved performance bounds in the context dimension d for offline and online learning.

Sparse Heterogeneity. Each problem instance (or, task) j is parameterized with a predictive parameter vector β^j — e.g., the parameters of a linear regression model predicting the reward of each decision as a function of the observed features. Without loss of generality, we can write

$$\beta^j = \beta^\dagger + \delta^j,$$

where β^\dagger represents the portion of the parameter vector that is “shared” across similar tasks, and δ^j is the task-specific portion that represents idiosyncratic biases specific to task j . Sparse heterogeneity imposes that the task-specific bias δ^j is *sparse* — i.e., only a few of its components are nonzero (see, e.g., Bastani 2021, Xu et al. 2021, Tian and Feng 2022, Li et al. 2023). Prior work has argued that this is the case when some (unknown) mechanism systematically affects a subset of the features, e.g., some hospitals under-diagnose certain conditions (Bastani 2021), or a domain change affects the meaning of a subset of words in natural language (Xu et al. 2021).

We empirically examine this assumption in the context of our previous Example 1. Specifically, we train separate linear models $\{\hat{\beta}_j\}_{j=1}^{13}$ for predicting diabetes risk at each of the 13 hospitals

using hospital-specific electronic medical record data. Then, we use the trimmed mean to estimate the shared model (for reasons explained in §3), and compute the resulting task-specific parameters $\{\hat{\delta}_j\}_{j=1}^{13}$ for each hospital. If there was no idiosyncratic task-specific bias for each hospital, these parameters would be statistically indistinguishable from zero; on the other hand, if the predictive tasks for each hospital had no shared structure, these parameters would be large and non-sparse. Each row of Figure 2 below shows a heatmap of the nonzero coefficients of the task-specific parameter $|\hat{\delta}_j|$ across 77 features used for prediction.¹ We find that each task exhibits statistically distinguishable hospital-specific idiosyncrasies in the underlying predictive model. Furthermore, in support of our hypothesis of sparse heterogeneity, each $\|\hat{\delta}_j\|_0 \leq 8 \ll 77$, i.e., task-specific parameters are s -sparse with $s/d \lesssim 0.1$ (see §6 for more details).

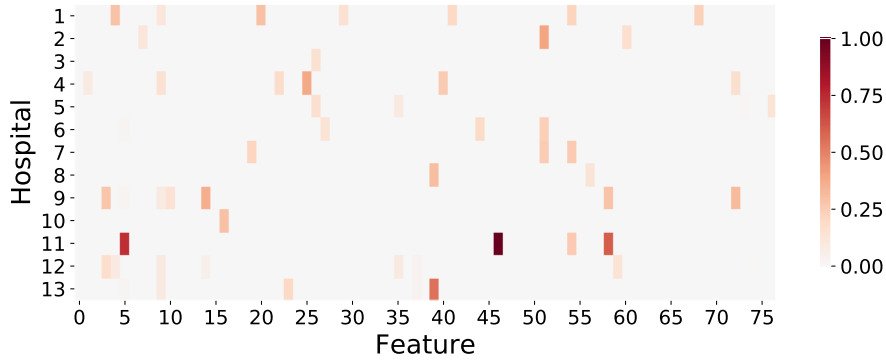


Figure 2 Heatmap of nonzero coefficients given by the estimated task-specific parameters $\{|\delta^j|\}_{j=1}^{13}$ for each hospital. Each row represents one of 13 hospitals and each column represents one of 77 features extracted from electronic medical record data. Nonzero coefficients are determined by a bootstrap hypothesis test across 500 random draws of the training data; we set coefficient i of row j to be zero if the null hypothesis ($\hat{\delta}_{(i)}^j = 0$) is not rejected at a 5% significance level. Each $\|\hat{\delta}^j\|_0 \ll d$, lending support for our hypothesis of sparse heterogeneity.

Existing multitask learning algorithms (e.g., pooling data or regularizing estimates across problem instances) are not designed to leverage this structure (see §1.1 for an overview of current methods). Thus, we first propose a novel two-stage multitask estimator, **RMEstimator**, that exploits this structure in the supervised learning setting. In the first stage, it leverages the trimmed mean from robust statistics (Rousseeuw 1991, Lugosi and Mendelson 2021) to estimate a “shared” model $\hat{\beta}^\dagger$ across data collected from similar tasks.² Then, in the second stage, it uses LASSO regression (Chen et al. 1995, Tibshirani 1996a) to efficiently learn the task-specific bias δ^j , which can be

¹ We use a bootstrap hypothesis test across 500 random draws of the training data to determine whether a coefficient is statistically distinguishable from zero (i.e., the null hypothesis $\hat{\delta}_{(i)}^j = 0$ is not rejected at the 5% significance level). These results are consistent across different hyperparameters used for the trimmed mean to determine $\hat{\beta}^\dagger$.

² Note that we do not attempt to estimate the original shared parameter β^\dagger , since it is not identifiable; rather, as discussed in §3, it suffices to estimate some $\tilde{\beta}^\dagger$ that lies in an ℓ_0 ball of radius $\mathcal{O}(s)$ around β^\dagger .

combined with our estimate of β^\dagger to obtain the task-specific parameter β^j . We prove finite-sample generalization bounds that show favorable performance (compared to popular baselines), especially in terms of the feature dimension d ; importantly, this error bound improvement is *exponential* for “data-poor” learning tasks, which stand to benefit the most from shared learning.

As noted earlier, we believe such a data-efficient approach is especially useful for bandit problems, since they often operate with limited historical data. To this end, we extend our results to online learning by embedding the **RMEstimator** within simultaneous linear contextual bandit algorithms running at each problem instance. We specify a dynamic calibration of our estimator within the **RMBandit** algorithm to appropriately balance the bias-variance tradeoff arising from incorporating auxiliary data from similar bandit instances (multitask learning) in conjunction with the classical exploration-exploitation tradeoff (bandit learning). We derive upper bounds for the cumulative regret of the **RMBandit**, demonstrating improvements in the context dimension d ; analogous to the offline setting, this regret improvement is exponential for data-poor bandit instances.

Finally, we empirically evaluate our approach on both synthetic and real datasets in healthcare and pricing. We find that our multitask learning strategy based on the **RMEstimator** substantially speeds up learning and improves overall performance in both offline and online settings.

1.1. Related Literature

Our work relates to the literature on multitask learning and contextual bandits; we contribute on both fronts. Our approach builds on the literature on robust and high-dimensional statistics.

There has been significant interest from the machine learning community on developing methods that combine data from multiple learning problems (typically referred to as tasks). These can be broadly classified into three categories: (i) multitask learning (Caruana 1997), where one aims to learn jointly across a fixed set of similar tasks, (ii) transfer learning (Pan et al. 2010), a special case of multitask learning, where the goal is to maximize performance on a distinguished “target” task, and (iii) meta-learning (Finn et al. 2017), where one aims to learn from historical tasks to improve learning in similar future tasks. Our problem is an instance of multitask learning, since our goal is to learn across a fixed set of problem instances with related unknown parameters.

Multitask Learning. Naturally, if the tasks are sufficiently different, then learning in one task cannot substantially improve learning in other tasks (Hanneke and Kpotufe 2020). Thus, a common approach in machine learning is to assume that the underlying parameters across tasks are close in ℓ_2 norm. Joint learning can then be operationalized by regularizing the estimated parameters together, e.g., through ridge (Evgeniou and Pontil 2004) or kernel ridge (Evgeniou et al. 2005) regularization. Alternatively, one can employ a shared Bayesian prior across tasks (Raina et al. 2006, Gupta and Kallus 2021) or simply pool data from nearby tasks (Ben-David et al. 2010,

Crammer et al. 2008). However, these approaches do not improve performance bounds beyond constants; in general, one must impose (and exploit) additional structure to obtain nontrivial theoretical improvements. Bastani (2021) uses real datasets to motivate the assumption that the parameters across tasks are close in ℓ_0 norm. This structure motivates a two-step estimator of transfer learning using LASSO regression, yielding improved bounds in the feature dimension d for supervised learning (Bastani 2021, Li et al. 2020a, Tian and Feng 2022, Li et al. 2023) and unsupervised learning (Xu et al. 2021). One can further impose that the underlying parameters for each task are sparse, sharing the same support (Lounici et al. 2009, Li et al. 2023) or similar covariance matrices (Li et al. 2020a, Tian and Feng 2022) across tasks; we do not make these assumptions since the applications we consider often have dense underlying parameters (see, e.g., Bastani 2021) and the covariance matrices vary widely across tasks due to covariate shifts (e.g., due to different customer populations at hospitals or stores, see Subbaswamy and Saria 2020).

We build on the last stream of two-step estimators for the multitask learning problem. However, we need a fundamentally different algorithmic approach; as we discuss in §3, the challenge is that the sparse bias terms can be poorly aligned across tasks, and thus classical estimates of the shared model (e.g., via data pooling or model averaging) destroy task-specific sparse structure and therefore cannot be debiased using LASSO (as was the case in prior work). Instead, we take the view that each component where the bias terms align poorly suffers “corruptions” to the shared model; we use a counting argument to show that either the number of corruptions must be small, or the component is one of a small number of well-aligned components. We use robust statistics to overcome corruptions for poorly-aligned components and LASSO to debias well-aligned components. To the best of our knowledge, our work proposes the first such combination of robust statistics and high-dimensional regression, yielding improved bounds for multitask learning.

The first step of our approach (using robust statistics) relates to recent robust machine learning methods that can handle adversarial corruptions to a small fraction of the data (Yin et al. 2018, Konstantinov and Lampert 2019). These approaches do not apply to our setting — as a consequence of our sparse differences assumption, we show that only a few similar *tasks* (as opposed to observations or features) have unknown parameters that are “corrupted” in most dimensions. Rather, we build on the classical trimmed mean estimator (Rousseeuw 1991, Lugosi and Mendelson 2021). The second step (using LASSO) builds on the high-dimensional statistics literature (Tibshirani 1996a, Candes and Tao 2007, Bickel et al. 2009, Bühlmann and Van De Geer 2011).

Multitask Bandits. A few recent papers have studied multitask learning across contextual bandit instances; however, to the best of our knowledge, a key drawback of these algorithms is that none of them ultimately improve the regret bounds for any bandit instance beyond constants. Similar to the multitask learning literature discussed above, one strategy is to regularize the learned parameters

for a given bandit instance towards parameters for similar bandit instances (Soare et al. 2014). For example, Cesa-Bianchi et al. (2013) and Deshmukh et al. (2017) leverage parameter updates that are similar to kernel ridge regularization, and Gentile et al. (2014) additionally perform a pre-processing step clustering bandit instances prior to such regularization; however, the resulting regret bound for a single bandit instance may actually *increase* in the number of instances N . Another popular approach is to impose a shared Bayesian prior across bandit instances (Cella et al. 2020, Bastani et al. 2021c, Kveton et al. 2021), but they also obtain similar results; furthermore, these algorithms require the more restrictive assumption that bandit instances appear sequentially (rather than simultaneously) in order to learn the prior. We embed our robust multitask estimator across N linear contextual bandit instances; the specific setting and assumptions we consider are based on Goldenshluger and Zeevi (2013), Bastani and Bayati (2020). We demonstrate that, unlike prior work, we obtain improved regret bounds for each bandit instance in the context dimension d under the practically-motivated sparse heterogeneity assumption; the improvement we obtain is exponential for data-poor instances where shared learning is most helpful.

We empirically illustrate the value of our approach on well-studied data-driven decision-making problems that leverage contextual bandit algorithms, such as personalized healthcare (Bastani and Bayati 2020, Zhalechian et al. 2022) and dynamic pricing (Ban and Keskin 2021, Wang et al. 2021).

1.2. Contributions

We highlight our main technical contributions below:

1. We introduce the **RMEstimator** for multitask learning, which leverages a unique combination of robust statistics (for learning a shared model across tasks) and LASSO (for debiasing this shared model for a specific task). We prove upper and lower bounds demonstrating that our estimator outperforms a number of intuitive baselines, and improves existing error bounds in the feature dimension d ; notably, this improvement is exponential for data-poor tasks.
2. We embed our estimator in a multitask contextual bandit framework and propose the **RMBandit** algorithm, with a suitable dynamic calibration of the **RMEstimator**. The resulting regret bounds analogously exhibit up to exponentially improved scaling in the context dimension d .

Finally, we conclude with numerical experiments on both synthetic and real datasets.

2. Problem Formulation

This section formulates multitask learning under sparse heterogeneity in the offline supervised learning setting; we extend our results to the online contextual bandit setting in §5.

Notation. Let $[n]$ denote the index set $\{1, 2, \dots, n\}$. For any vector $\beta \in \mathbb{R}^d$ and $i \in [d]$, let $\beta_{(i)}$ be the i^{th} element of β ; for any index set $\mathcal{I} \subseteq [d]$, let $\beta_{\mathcal{I}}$ denote the vector obtained by replacing the elements of β that are not in \mathcal{I} with 0s. We use superscripts to index the task, e.g., the design matrix \mathbf{X}^j represents the covariates observed at task j . Further, for any square matrix $\Sigma \in \mathbb{R}^{d \times d}$, let $\lambda_{\min}(\Sigma)$ and $\lambda_{\max}(\Sigma)$ denote its minimum and maximum eigenvalues respectively. We use the subscript (i, \cdot) to index the i^{th} row of a matrix, (\cdot, j) to index the j^{th} column, and (i, j) to index the $(i, j)^{\text{th}}$ element, e.g., $\mathbf{X}_{(i, \cdot)}$ is the i^{th} row of matrix \mathbf{X} .

Model. We consider N distinct problem instances, each facing a linear learning task, e.g., N hospitals in Example 1 or N stores in Example 2 respectively. Task $j \in [N]$ has n_j observations (e.g., patients or customers); an observation i is associated with a d -dimensional feature vector $X_i \in \mathbb{R}^d$. The response Y_i of an individual i from task j as

$$Y_i = X_i^\top \beta^j + \epsilon_i,$$

where the noise ϵ_i is an independent σ_j -subgaussian random variable (see Definition 1); note that the variance of the noise term can be task-dependent. Let the vector $Y^j \in \mathbb{R}^{n_j}$ encode all observed responses in task j , and the vector $\epsilon^j \in \mathbb{R}^{n_j}$ encode the corresponding noise terms.

DEFINITION 1. A random variable $Z \in \mathbb{R}$ with mean $\mu = \mathbb{E}[Z]$ is σ -subgaussian if, for any $\lambda \in \mathbb{R}$, $\mathbb{E}[\exp(\lambda(Z - \mu))] \leq \exp(\sigma^2 \lambda^2 / 2)$.

The formulation above captures any N linear instances; we now impose our assumption on sparse heterogeneity. As discussed in the introduction, we impose that each task's predictive parameter can be decomposed into a shared parameter β^\dagger (that captures the similarity across all N instances) and a task-specific parameter δ^j (that captures idiosyncratic behavior inherent to task j):

$$\beta^j = \beta^\dagger + \delta^j,$$

where δ^j is sparse (i.e., $\|\delta^j\|_0 \leq s$ for some $s \in \mathbb{N}$) for all $j \in [N]$. This key assumption enables us to learn across instances efficiently. Note that we do *not* assume that the individual arm parameters $\{\beta^j\}_{j \in [N]}$ or the shared models β^\dagger are themselves sparse, since the responses can often depend on the entire set of observed covariates (see, e.g., discussion in Bastani 2021).

REMARK 1. Note that the choice of the shared vector β^\dagger here is not unique — e.g., changes up to $\mathcal{O}(s)$ components of β^\dagger preserve the sparsity of δ^j up to constant factors — and therefore is not identifiable. As we describe in §3, it suffices for our purposes to estimate any vector $\tilde{\beta}^\dagger$ that lies in an $\mathcal{O}(s)$ ball in ℓ_0 norm centered around an admissible choice of β^\dagger .

Design Matrix. For each task j , we construct the usual design matrix $\mathbf{X}^j \in \mathbb{R}^{n_j \times d}$, where the i^{th} row $\mathbf{X}_{(i,\cdot)}^j = X_i$. Following standard practice for regularized regression (Tibshirani 1996b, Hastie et al. 2009), we standardize each feature such that each column i of the design matrices satisfies

$$\frac{1}{n_j} \|\mathbf{X}_{(\cdot,i)}^j\|_2^2 = 1. \quad (1)$$

We further define the corresponding sample covariance matrices as

$$\hat{\Sigma}^j = \frac{\mathbf{X}^{j\top} \mathbf{X}^j}{n_j}.$$

Due to our normalization, every entry on the diagonal of $\hat{\Sigma}^j$ is 1.

Performance. Our goal is to use the observed data $\{(\mathbf{X}^j, Y^j)\}_{j \in [N]}$ to estimate the unknown parameter vectors $\{\beta^j\}_{j \in [N]}$ for all tasks. We measure the performance of an estimator $\hat{\beta}^j$ by its ℓ_1 error, i.e., $\|\hat{\beta}^j - \beta^j\|_1$; a good estimator $\hat{\beta}^j$ has a small estimator error, and hence a small prediction error by noting that $|X_i^\top (\hat{\beta}^j - \beta^j)| \leq \|X_i\|_\infty \|\hat{\beta}^j - \beta^j\|_1$.

We first analyze the *fixed design* setting where the observations \mathbf{X}^j are treated as given (§4.1). We then show how our results straightforwardly extend to the *random design* setting where X_i is drawn i.i.d. from some (potentially unknown) distribution \mathcal{P}_X^j (§4.2). Note that the feature distribution can vary as a function of the task j ; this is because, in practice, different providers serve different populations, which is reflected in their feature vector distributions.

Regimes of Interest. While our primary result on the performance of the **RMEstimator** (Theorem 1 in §4.1) is general with respect to the sample sizes $\{n_j\}_{j=1}^N$, we will find it useful to interpret the implications under two intuitive regimes. The first is the “standard” regime where all tasks have roughly similar numbers of observations (i.e., $n_j = \Theta(n_{j'}) = \Theta(n/N)$ for any $j, j' \in [N]$, where $n = \sum_{j \in [N]} n_j$). However, in some settings, some service providers may receive substantially less traffic than others (e.g., a rural hospital in Example 1 or a relatively small store in Example 2). Thus, we also consider the limit where some task $j \in [N]$ is relatively “data-poor”, receiving far less traffic than other tasks (i.e., $n_j = \Theta(n'_j/d^2)$ for any $j' \neq j$); we refer to this as the “data-poor” setting. We focus on a single data-poor task for simplicity; our results generalize straightforwardly to the case where there are a constant number of data-poor tasks.

3. Robust Multitask Learning

In this section, we overview our robust multitask estimator **RMEstimator** (§3.2) and provide intuition for its design (§3.3). §3.4 illustrates the statistical benefits of our approach relative to intuitive baselines.

3.1. Preliminaries

We define and briefly review the *trimmed mean* estimator from the classical robust statistics literature (Rousseeuw 1991, Lugosi and Mendelson 2021), which computes the mean of a distribution \mathcal{P} given samples $\{Z_j\}_{j \in [N]}$. A typical setting is as follows: most of the samples are i.i.d. (i.e., $Z_j \sim \mathcal{P}$), but a small fraction (indexed by an unknown set $\mathcal{J} \subseteq [N]$) are “corrupted” and can be arbitrary. Here, the traditional mean can be arbitrarily biased, but the trimmed mean obtains strong guarantees given a bound on the number of corrupted samples $|\mathcal{J}| < \zeta N$ for some $\zeta < 1/2$. The trimmed mean estimator first sorts the samples in increasing order to obtain $Z_{j_1} \leq \dots \leq Z_{j_N}$, where the subscript j_i is the index of the i^{th} smallest sample. Then, given a hyperparameter $\omega > \zeta$, it removes the top and bottom ω quantiles or $N\omega$ values and takes the mean of the remaining ones — i.e.,

$$\text{TrimmedMean}(\{Z_j\}_{j \in [N]}, \omega) = \frac{1}{N(1-2\omega)} \sum_{i=N\omega+1}^{N(1-\omega)} Z_{j_i}.$$

Intuitively, this estimator is robust since either the corruptions are among the deleted values, or they are sufficiently close to the true mean that they do not significantly affect the estimate.

3.2. Algorithm Description

Our robust multitask estimator is summarized in Algorithm 1. At a high level, the first step combines high-variance ordinary least squares (OLS) estimators across instances using robust statistics to estimate the shared parameter β^\dagger (up to $\mathcal{O}(s)$ deviations in ℓ_0 norm); then, the second step uses LASSO regression to debias this estimate for each task $j \in [N]$.

Algorithm 1 Robust Multitask Estimator (RMEstimator)

Input parameters: λ, ω

Initialize $\lambda_j = \lambda / \sqrt{n_j}$

for $j \in [N]$ **do**

Let $\hat{\beta}_{\text{ind}}^j = (\mathbf{X}^{j\top} \mathbf{X}^j)^{-1} \mathbf{X}^j \mathbf{Y}^j$ be the OLS estimator for task j

end for

for $i \in [d]$ **do**

Let $\hat{\beta}_{\text{RM},(i)}^\dagger = \text{TrimmedMean}(\{\hat{\beta}_{\text{ind},(i)}^j\}_{j \in [N]}, \omega)$ be the element-wise trimmed mean

end for

for $j \in [N]$ **do**

Compute $\hat{\beta}_{\text{RM}}^j = \arg \min_{\beta} \left\{ \frac{1}{n_j} \|\mathbf{X}^j \beta - \mathbf{Y}^j\|_2^2 + \lambda_j \|\beta - \hat{\beta}_{\text{RM}}^\dagger\|_1 \right\}$

end for

Outputs: $\{\hat{\beta}_{\text{RM}}^j\}_{j \in [N]}$

In more detail,

- **Step 1 (Estimating β^\dagger):** We compute the usual OLS estimator

$$\hat{\beta}_{\text{ind}}^j = (\mathbf{X}^{j\top} \mathbf{X}^j)^{-1} \mathbf{X}^{j\top} Y^j$$

for each task $j \in [N]$ independently. Then, we combine these estimates using the element-wise trimmed mean to estimate the shared parameter vector $\hat{\beta}_{\text{RM}}^\dagger \approx \beta^\dagger$ — i.e., for each $i \in [d]$,

$$\hat{\beta}_{\text{RM},(i)}^\dagger = \text{TrimmedMean} \left(\{\hat{\beta}_{\text{ind},(i)}^j\}_{j \in [N]}, \omega \right), \quad (2)$$

where $\omega > 0$ is the trimming hyperparameter that we specify later. (Recall that the subscript (i) represents the i^{th} entry of the vector.)

- **Step 2 (Estimating β^j):** Next, we use LASSO regression to compute $\hat{\beta}_{\text{RM}}^j$, leveraging our assumption that the instance-specific bias term $\beta^j - \beta^\dagger$ is sparse:

$$\hat{\beta}_{\text{RM}}^j = \arg \min_{\beta} \left\{ \frac{1}{n_j} \|\mathbf{X}^j \beta - Y^j\|_2^2 + \lambda_j \|\beta - \hat{\beta}_{\text{RM}}^\dagger\|_1 \right\} \quad (3)$$

3.3. Design Intuition

We now provide intuition for our design choices relative to alternative strategies; the corresponding error rates are summarized in Table 1 (see §3.4 for precise definitions and more details).

Estimator	Estimation Error		Bound Type
	<i>Standard Regime</i>	<i>Data-Poor Regime</i>	
Independent $\hat{\beta}_{\text{ind}}^j$	$\frac{d}{\sqrt{n_j}}$	$\frac{d}{\sqrt{n_j}}$	Lower
Averaging $\hat{\beta}_{\text{avg}}^j$ or Pooling $\hat{\beta}_{\text{pool}}^j$	$\ \delta^j\ _1 + \frac{d}{\sqrt{N n_j}}$	$\ \delta^j\ _1 + \frac{1}{\sqrt{N n_j}}$	Lower
Averaging Multitask $\hat{\beta}_{\text{AM}}^j$	$\frac{\min\{Ns, d\}}{\sqrt{n_j}} + \frac{d}{\sqrt{N n_j}}$	$\frac{\min\{Ns, d\}}{\sqrt{n_j}}$	Lower
Robust Multitask $\hat{\beta}_{\text{RM}}^j$	$\sqrt{\frac{sd}{n_j}} + \frac{d}{\sqrt{N n_j}}$	$\frac{s}{\sqrt{n_j}}$	Upper

Table 1 Comparison of parameter estimation error $\sup_{\mathcal{G}} \mathbb{E} [\|\hat{\beta}^j - \beta^j\|_1]$ (see §3.4 for the precise definitions of these estimators); constants and logarithmic factors are omitted for clarity. The upper bound for our robust multitask estimator outperforms the worst-case lower bounds for intuitive baseline estimators under the same set of problem settings \mathcal{G} ; our improvement is largest for data-poor instances.

One strategy is to simply use the *independent* OLS estimator $\hat{\beta}_{\text{ind}}^j$ (from Step 1) to estimate β^j ; this is an unbiased estimator, but has very high variance since it only uses the limited data observed in task j and does not leverage shared structure across instances. As a result, it has high error when n_j is small (see Table 1).

An alternative strategy is to estimate the shared model β^\dagger using data across instances, e.g., the *averaging* estimator takes the model average of the independent estimators:

$$\hat{\beta}_{\text{avg}}^j = \frac{1}{N} \sum_{i \in [N]} \hat{\beta}_{\text{ind}}^i.$$

This estimator has low variance since it leverages data across tasks, but it is biased since it does not account for the task-specific idiosyncratic bias term $\delta^j = \beta^j - \beta^\dagger$. Similarly, estimating the shared model β^\dagger through OLS on data *pooled* across instances suffers the same drawbacks. As shown in Table 1, the error of such estimators never approaches zero due to the bias term δ^j .

Thus, a natural two-step strategy to achieve low variance and low bias is to first compute an estimate $\hat{\beta}^\dagger$ of the shared parameter, and then try to *debias* it to estimate β^j . Since the bias $\beta^j - \beta^\dagger$ is s -sparse by assumption, it should intuitively be easier to debias $\hat{\beta}^\dagger$ than to directly estimate β^j .

Along these lines, consider the following *averaging multitask* estimator, denoted by the subscript AM. Here, we estimate the shared parameter via model averaging, $\hat{\beta}_{\text{AM}}^\dagger = \hat{\beta}_{\text{avg}}^j$. Then, we use an ℓ_1 penalty on $\beta - \hat{\beta}_{\text{AM}}^\dagger$ (i.e., LASSO regression) on data from instance j to debias $\hat{\beta}_{\text{AM}}^\dagger$:

$$\hat{\beta}_{\text{AM}}^j = \arg \min_{\beta} \left\{ \frac{1}{n_j} \|\mathbf{X}^j \beta - Y^j\|_2^2 + \lambda_j \|\beta - \hat{\beta}_{\text{AM}}^\dagger\|_1 \right\}. \quad (4)$$

(Note that this strategy is identical to Algorithm 1, except it uses the traditional mean instead of the trimmed mean in Step 1.) To see why equation (4) helps, suppose we had a perfect estimate of the shared model $\hat{\beta}_{\text{AM}}^\dagger = \beta^\dagger$; then, $\beta^j - \hat{\beta}_{\text{AM}}^\dagger$ would be s -sparse, in which case LASSO requires exponentially fewer observations for recovering β^j (relative to $\hat{\beta}_{\text{AM}}^\dagger$) than traditional OLS.

The issue with the approach outlined above is that $\beta^j - \hat{\beta}_{\text{AM}}^\dagger$ is *not* s -sparse, or even “close” to being s -sparse. To illustrate, we can decompose

$$\beta^j - \hat{\beta}_{\text{AM}}^\dagger = \underbrace{\beta^j - \beta^\dagger}_{s\text{-sparse}} + \underbrace{\beta^\dagger - \tilde{\beta}_{\text{AM}}^\dagger}_{(Ns)\text{-sparse}} + \underbrace{\tilde{\beta}_{\text{AM}}^\dagger - \hat{\beta}_{\text{AM}}^\dagger}_{\text{not sparse but small}}, \quad \text{where } \tilde{\beta}_{\text{AM}}^\dagger = \frac{1}{N} \sum_{j \in [N]} \beta^j \quad (5)$$

Here, $\tilde{\beta}_{\text{AM}}^\dagger$ is the value that $\hat{\beta}_{\text{AM}}^\dagger$ converges to as $n_j \rightarrow \infty$ for all $j \in [N]$. Note that $\tilde{\beta}_{\text{AM}}^\dagger$ does not converge to β^\dagger ; in fact, as noted in the problem formulation, β^\dagger is not identifiable. The first term in the decomposition is sparse, and the third term becomes small as $n = \sum_{j \in [N]} n_j$ becomes large (since $\hat{\beta}_{\text{AM}}^\dagger$ effectively uses all n samples to estimate $\tilde{\beta}_{\text{AM}}^\dagger$); since LASSO can effectively recover parameters that are approximately sparse, these two terms are not problematic. The key issue is the second term:

$$\tilde{\delta}_{\text{AM}}^\dagger = \tilde{\beta}_{\text{AM}}^\dagger - \beta^\dagger = \frac{1}{N} \sum_{j \in [N]} (\beta^j - \beta^\dagger) = \frac{1}{N} \sum_{j \in [N]} \delta^j, \quad (6)$$

which is neither sparse nor small. This is illustrated in Figure 3: since the support of the different bias terms $\{\delta_i\}_{i \in [N]}$ can be “poorly-aligned” (i.e., the idiosyncrasies for each task affect a different subset of features), the average across instances can result in $\tilde{\delta}_{\text{AM}}^\dagger$ having as many as $\min\{Ns, d\}$ nonzero components (even as $n_j \rightarrow \infty$ for all $j \in [N]$). This in turn implies that $\beta^j - \hat{\beta}_{\text{AM}}^\dagger$ is not sparse even for moderate values of N such as $N = \Omega(d/s)$; thus, we cannot use LASSO to efficiently debias $\hat{\beta}_{\text{AM}}^\dagger$. Other classical estimators of the shared parameter (e.g., data pooling) suffer the same issue.

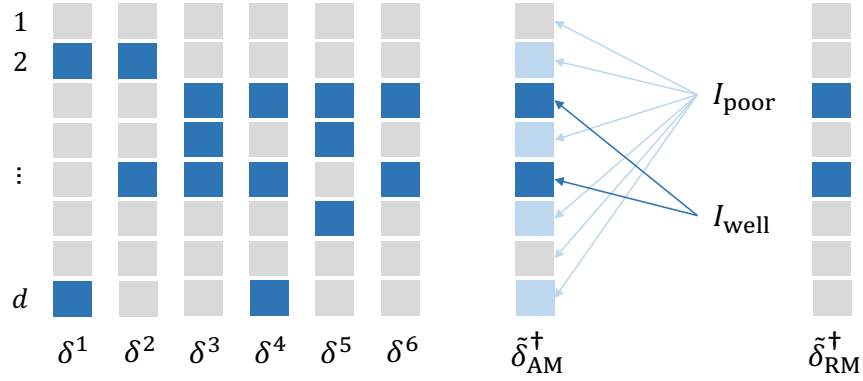


Figure 3 Illustration of Step 1 of our robust multitask estimator for debiasing data collected from multiple instances. Blue squares depict the support; the shade of blue depicts the magnitude. $\mathcal{I}_{\text{poor}}$ represents the index set which can be debiased using the trimmed mean across instances, while $\mathcal{I}_{\text{well}}$ represents the index set which can be debiased using a subsequent LASSO regression for the target instance.

REMARK 2. Recall Figure 2, mapping the estimated bias terms $\{\hat{\delta}_i\}_{i \in [N]}$ from electronic medical record data for diabetes prediction (Example 1). Indeed, we observe that $\{\hat{\delta}_i\}_{i \in [N]}$ are “poorly-aligned” (i.e., share support on different subsets of features), similar to the illustration in Figure 3.

Our *robust multitask* estimator addresses this issue by using the trimmed mean $\hat{\beta}_{\text{RM}}^\dagger$ in Step 1; we will show that this converges to a value $\tilde{\beta}_{\text{RM}}^\dagger$ (as $n_j \rightarrow \infty$ for all $j \in [N]$) such that

$$\tilde{\delta}_{\text{RM}}^\dagger = \tilde{\beta}_{\text{RM}}^\dagger - \beta^\dagger = \text{TrimmedMean}(\{\beta^j\}_{j \in [N]} - \beta^\dagger, \omega) = \text{TrimmedMean}(\{\delta^j\}_{j \in [N]}, \omega)$$

is $\mathcal{O}(s)$ -sparse. In particular, we have the following decomposition:

$$\beta^j - \hat{\beta}_{\text{RM}}^\dagger = \underbrace{\beta^j - \beta^\dagger}_{s\text{-sparse}} + \underbrace{\beta^\dagger - \tilde{\beta}_{\text{RM}}^\dagger}_{\mathcal{O}(s)\text{-sparse}} + \underbrace{\tilde{\beta}_{\text{RM}}^\dagger - \hat{\beta}_{\text{RM}}^\dagger}_{\text{not sparse but small}}. \quad (7)$$

As discussed above, the third term becomes small as n becomes large. Since the second term $\tilde{\delta}_{\text{RM}}^\dagger$ is $\mathcal{O}(s)$ -sparse, $\beta^j - \hat{\beta}_{\text{RM}}^\dagger$ is approximately $\mathcal{O}(s)$ -sparse; thus, LASSO can efficiently debias $\hat{\beta}_{\text{RM}}^\dagger$.

We now use a counting argument to illustrate why $\tilde{\delta}_{\text{RM}}^\dagger$ is $\mathcal{O}(s)$ -sparse. As Figure 3 illustrates, we can separate the components $i \in [d]$ into two groups: ones that are “well-aligned” ($i \in \mathcal{I}_{\text{well}}$) and ones that are “poorly-aligned” ($i \in \mathcal{I}_{\text{poor}}$); see Definition 2 below. A poorly-aligned component i is one where very few tasks $j \in [N]$ are biased in this component, i.e., $\beta_{(i)}^j \neq \beta_{(i)}^\dagger$. Intuitively, for each such component, the trimmed mean estimator treats these biased tasks as “corruptions” to our samples $\{\beta_{(i)}^j\}_{j \in [N]}$, and trims them (with high probability) when computing the average to obtain an unbiased estimate of $\beta_{(i)}^\dagger$. On the other hand, well-aligned components may remain arbitrarily biased. However, the pigeonhole principle implies that there cannot be many well-aligned components; thus, these components (in addition to the components affected by the sparse task-specific bias term) can be efficiently debiased by LASSO in Step 2. We now formalize this.

DEFINITION 2 (WELL- AND POORLY-ALIGNED COMPONENTS). Given a constant $\zeta \in [0, 1]$, a component $i \in [d]$ is ζ -*poorly-aligned* (denoted $i \in \mathcal{I}_{\text{poor}}^\zeta$) if

$$\frac{|\{j \in [N] \mid \beta_{(i)}^j \neq \beta_{(i)}^\dagger\}|}{N} < \zeta.$$

Otherwise, it is ζ -*well-aligned* (denoted $i \in \mathcal{I}_{\text{well}}^\zeta$).

In other words, a component i is ζ -poorly-aligned if at most ζ fraction of the N tasks satisfy $\beta_{(i)}^j \neq \beta_{(i)}^\dagger$. Now, Step 1 constructs an estimator $\hat{\beta}_{\text{RM}}^\dagger$ of β^\dagger that converges to

$$\tilde{\beta}_{\text{RM},(i)}^\dagger = \begin{cases} \beta_{(i)}^\dagger & \text{if } i \in \mathcal{I}_{\text{poor}}^\zeta \\ \text{unspecified} & \text{if } i \in \mathcal{I}_{\text{well}}^\zeta \end{cases}$$

as the sample sizes $\{n_j\}_{j \in [N]}$ become large. That is, we aim to correctly estimate all the poorly-aligned components, but the well-aligned components can be anything. We estimate each component $\beta_{(i)}^\dagger$ using the trimmed mean, which is robust to a small fraction ζ of arbitrarily corrupted samples. For a given component i , let the corresponding corrupted tasks be

$$\mathcal{J}_i = \{j \in [N] \mid \beta_{(i)}^j \neq \beta_{(i)}^\dagger\}.$$

By definition, for $i \in \mathcal{I}_{\text{poor}}^\zeta$, we have $|\mathcal{J}_i| < N\zeta$. Thus, we can use the trimmed mean estimator to estimate $\beta_{(i)}^\dagger$:

$$\hat{\beta}_{\text{RM},(i)}^\dagger = \text{TrimmedMean}\left(\{\hat{\beta}_{\text{ind},(i)}^j\}_{j \in [N]}, \omega\right)$$

for some $\omega > \zeta$. This strategy ensures that $\hat{\beta}_{\text{RM},(i)}^\dagger \approx \beta_{(i)}^\dagger$ for each poorly-aligned component as desired. Now, note that there can only be a few well-aligned components. In particular, out of the Nd total components in $\{\beta_{(i)}^j\}_{j \in [N]}$, there are at most Ns components where $\beta_{(i)}^j \neq \beta_{(i)}^\dagger$ as a consequence of our sparse heterogeneity assumption. Then, by the pigeonhole principle, we have

$$|\mathcal{I}_{\text{well}}^\zeta| \leq \frac{Ns}{N\zeta} = \frac{s}{\zeta}.$$

In other words, there are at most s/ζ well-aligned components, so $\tilde{\delta}_{\text{RM}}^\dagger$ is $\mathcal{O}(s)$ -sparse as desired (for a constant choice of ζ). Thus, we can efficiently debias our estimate using LASSO in Step 2.

3.4. Comparison with Baselines

With the intuition from §3.3 in hand, we now formalize the results for the baseline estimators in Table 1. In particular, we contrast the *upper bound* of our estimator (that we will derive in §4.1) with *lower bounds* of these baselines. Full statements and proofs are provided in Appendix B.

We characterize the estimation error of an estimator $\hat{\beta}^j$ through the following loss function:

$$\ell(\hat{\beta}^j, \beta^j) = \sup_{\mathcal{G}} \mathbb{E} \left[\|\hat{\beta}^j - \beta^j\|_1 \right], \quad (8)$$

where $\mathcal{G} = \{\{\mathbf{X}^j\}_{j \in [N]}, \{\beta^j\}_{j \in [N]}, \{\mathcal{P}_\epsilon^j\}_{j \in [N]}\}$ is the set of problem instances that satisfies our fixed design formulation in §2 and has positive-definite sample covariance matrices $\{\hat{\Sigma}^j\}_{j \in [N]}$ (see Assumption 1 in §4.1), ensuring that the independent OLS estimators are well-defined. The expectation is taken with respect to the distribution \mathcal{P}_ϵ^j of the subgaussian noise terms ϵ^j . This choice of \mathcal{G} ensures that our upper and lower bounds are with respect to the same class of problem instances.

We consider the following estimators:

- **Independent OLS:** This is the OLS estimator $\hat{\beta}_{\text{ind}}^j = (\mathbf{X}^{j\top} \mathbf{X}^j)^{-1} \mathbf{X}^{j\top} Y^j$ trained on data only from task j , i.e., it does not transfer information across tasks.

PROPOSITION 1. *The estimation error of the independent estimator satisfies*

$$\ell(\hat{\beta}_{\text{ind}}^j, \beta^j) = \Omega\left(\frac{d}{\sqrt{n_j}}\right).$$

- **Averaging/Pooling:** The averaging estimator $\hat{\beta}_{\text{avg}}^j = \frac{1}{N} \sum_{i \in [N]} \hat{\beta}_{\text{ind}}^i$ is a common approach that averages the independent OLS estimates across tasks (see, e.g., Dobriban and Sheng 2021); the pooling estimator $\hat{\beta}_{\text{pool}}^j = \left(\sum_{i \in [N]} \mathbf{X}^{i\top} \mathbf{X}^i\right)^{-1} \left(\sum_{i \in [N]} \mathbf{X}^{i\top} Y^i\right)$ pools data across tasks to train a single OLS estimator (see, e.g., Crammer et al. 2008, Ben-David et al. 2010).

PROPOSITION 2. *The estimation error of the averaging/pooling estimator satisfies*

$$\ell(\hat{\beta}_{\text{avg}}^j, \beta^j) = \Omega\left(\|\delta^j\|_1 + \frac{d}{\sqrt{Nn_j}}\right).$$

- **Averaging Multitask:** This two-step estimator $\hat{\beta}_{\text{AM}}^j$ is described in detail in §3.3. It is an ablation of our robust multitask estimator that uses the traditional mean rather than the trimmed mean in Step 1. The proof follows a LASSO lower bound argument as in Theorem 7.1 of Lounici et al. (2011), where λ_j takes the value chosen to upper bound the estimation error.

PROPOSITION 3. *Let the hyperparameter $\lambda_j = \sqrt{\frac{32(1+C)\sigma_j^2 \log(4d)}{n_j}}$ for some constant $C \geq 3$. Then, the estimation error of the averaging multitask estimator satisfies*

$$\ell(\hat{\beta}_{\text{AM}}^j, \beta^j) = \tilde{\Omega}\left(\frac{\min\{Ns, d\}}{\sqrt{n_j}} + \frac{d}{\sqrt{Nn_j}}\right).$$

As shown in Table 1 (and proven in the next section), the upper bound of our RMEstimator outperforms the lower bounds of these estimators by leveraging sparse heterogeneity.

4. Theoretical Guarantees for Multitask Learning

Next, we prove performance guarantees for the RMEstimator under both fixed (§4.1) and random (§4.2) design; the statistical benefits of our approach are magnified for data-poor tasks (§4.3). §4.4 provides guidance on choosing a subset of tasks on which to perform shared learning.

4.1. Main Result

Our fixed design result holds under the standard assumption that the sample covariance matrices $\{\widehat{\Sigma}^j\}_{j \in [N]}$ are positive-definite (Hastie et al. 2009), i.e., the OLS estimator is well-defined for each task. (We will subsequently relax this assumption for random designs and for data-poor tasks.)

ASSUMPTION 1 (Positive-Definiteness). *There exists a constant $\psi > 0$ such that for any $j \in [N]$ we have $\lambda_{\min}(\widehat{\Sigma}^j) \geq \psi$.*

We first show an intermediate result on the error of the trimmed mean estimator (Stage 1 of the RMEstimator) for N data samples, of which ζ fraction may be arbitrarily corrupted.

PROPOSITION 4. *Suppose we are given N samples $\{Z_j\}_{j \in [N]}$ and a subset $\mathcal{J} \subseteq [N]$ of size $|\mathcal{J}| < \zeta N$ with $\zeta \in (0, 1/2)$, such that $\{Z_j\}_{j \in \mathcal{J}^c}$ are independent σ_j -subgaussian random variables with equal means $\mu = \mathbb{E}[Z_j]$ and $\{Z_j\}_{j \in \mathcal{J}}$ can be arbitrarily corrupted. Then, letting $\widehat{\mu} = \text{TrimmedMean}(\{Z_j\}_{j \in [N]}, \omega)$ with $\omega = \zeta + \eta$, we have*

$$\mathbb{P} \left[|\widehat{\mu} - \mu| \geq C_0 \max_{j \in \mathcal{J}^c} \sigma_j (3\zeta + 4\eta) \sqrt{\log\left(\frac{3}{\eta}\right)} \right] \leq 3 \exp\left(-\frac{N\eta^2}{9}\right),$$

for any $0 < \eta \leq 1/2 - 1/C_0 - \zeta$ with some constant $C_0 > 2$.

The proof is provided in Appendix A.1. We use this result to show that $\widehat{\beta}_{\text{RM},(i)}^\dagger$ is close to the true mean $\beta_{(i)}^\dagger$ for poorly-aligned components $i \in \mathcal{I}_{\text{poor}}^\zeta$. This result is similar to classical results from robust statistics (see, e.g., Li 2019), but existing results typically assume that the uncorrupted samples are i.i.d., whereas we only require independence (since we wish to apply it to $\{\widehat{\beta}_{\text{ind},(i)}^j\}_{j \in [N]}$, which might not be identically distributed).

As described in Eq. (7) in §3.3, the remainder term $\beta^j - \widehat{\beta}_{\text{RM}}^\dagger$ is approximately sparse, allowing us to use the LASSO estimator (Stage 2 of the RMEstimator) to efficiently recover each task-specific β^j . This yields our main error bound for the RMEstimator:

THEOREM 1. *Under Assumption 1, the estimator $\widehat{\beta}_{\text{RM}}^j$ with hyperparameter $\zeta \in (0, 1/2)$ satisfies*

$$\|\widehat{\beta}_{\text{RM}}^j - \beta^j\|_1 \leq \frac{6\lambda_j s}{\zeta\psi} + C_0 d (3\zeta + 4\eta) \max_{i \in [N]} \sqrt{\frac{\sigma_i^2}{n_i\psi} \log\left(\frac{3}{\eta}\right)}$$

with at least a probability of $1 - \left(3d \exp\left(-\frac{N\eta^2}{9}\right) + 2d \exp\left(-\frac{\lambda_j^2 n_j}{32\sigma_j^2}\right)\right)$, for any $\lambda_j > 0$ and $0 < \eta \leq 1/2 - 1/C_0 - \zeta$ with some constant $C_0 > 2$.

The proof is provided in Appendix A.2. Theorem 1 holds generally for any choice of regularization parameter λ_j , but we can choose it to minimize the error bound.

To better interpret the resulting implications, as discussed in § 2, we consider a “standard” regime, where all tasks have roughly similar sample sizes, i.e., $n_j = \Theta(n_{j'}) = \Theta(n/N)$ for any $j, j' \in [N]$, where $n = \sum_{j \in [N]} n_j$. Under this regime, the following corollary is a direct application of Theorem 1, with the choice of λ_j provided in Algorithm 1.

COROLLARY 1. *Under the standard regime and Assumption 1, the estimator $\widehat{\beta}_{RM}^j$ satisfies*

$$\|\widehat{\beta}_{RM}^j - \beta^j\|_1 = \tilde{O}\left(\sqrt{\frac{sd}{n_j}} + \frac{d}{\sqrt{Nn_j}}\right),$$

with at least a probability of $1 - \delta$ for any $\delta \geq \exp\left(-\frac{N}{9}\left(\frac{C_0-2}{4C_0}\right)^2 + \log(6d)\right)$, for appropriate choices of hyperparameters ζ , η , and λ provided in Appendix A.3.

The proof is provided in Appendix A.3. It is useful to compare the bound above with that of a single task j in the same setting, but which does *not* leverage knowledge sharing with other simultaneous tasks. Recall that the independent OLS estimator $\widehat{\beta}_{\text{ind}}^j$ on task j yields an estimation error of $\mathcal{O}(d/\sqrt{n_j})$. In contrast, if the number of tasks is at least $N = \Omega(d/s)$, our robust multitask estimator has an estimation error of at most $\tilde{O}\left(\sqrt{sd/n_j}\right)$ with high probability, i.e., it improves the upper bound by a factor of \sqrt{d} . This can be a substantial improvement in high dimension (large d) and underscores the value of learning across tasks. When we have very few tasks from which to share knowledge (i.e., $N = o(d)$), multitask learning is still effective; in the worst case, we obtain the same estimation error as the independent OLS estimator. As we show in Section 6, we obtain improved empirical results in practice even for modest values of N . As we will discuss in §4.3, our improvement is much larger in the data-poor regime.

4.2. Random Design

Our previous results were under a fixed design — i.e., the $\{\mathbf{X}^j\}_{j \in [N]}$ are observed. As discussed in §2, our results straightforwardly extend to random designs, where each row of \mathbf{X}^j is randomly drawn from a distribution \mathcal{P}_X^j . Now, instead of assuming positive-definiteness for each sample covariance matrix (Assumption 1), it suffices to only assume it for each *true* covariance matrix.

ASSUMPTION 2 (**Positive-Definiteness**). *There exists a constant $\tilde{\psi} > 0$ such that for any $j \in [N]$ we have $\lambda_{\min}(\Sigma^j) \geq \tilde{\psi}$.*

Note that, under a random design, we can no longer standardize the features as in Eq. (1). Thus, we also assume that the observed features are bounded.

ASSUMPTION 3 (**Boundedness**). *There exists a constant $x_{\max} > 0$ such that $\|X_i\|_{\infty} \leq x_{\max}$.*

REMARK 3. The literature on OLS/ridge regression typically assumes a bound on the ℓ_2 -norm of the covariate X_i (see, e.g., Hsu et al. 2011, 2012, Zhang 2005, Smale and Zhou 2007). This kind of assumption can be relaxed using moment or subgaussian conditions (see, e.g., Condition 3 in Hsu et al. 2011) at the cost of an extra term in the resulting high probability bound.

Now we introduce a variant of Theorem 1 below for the random design setting.

PROPOSITION 5. Under Assumptions 2 and 3, the estimator $\widehat{\beta}_{RM}^j$ with hyperparameter $\zeta \in (0, 1/2)$ satisfies

$$\|\widehat{\beta}_{RM}^j - \beta^j\|_1 \leq \frac{12\lambda_j s}{\zeta\psi} + C_0 d(3\zeta + 4\eta) \max_{i \in [N]} \sqrt{\frac{2\sigma_i^2}{n_i\psi} \log\left(\frac{3}{\eta}\right)}$$

with at least a probability of $1 - \left(3d \exp(-\frac{N\eta^2}{9}) + 2d \exp(-\frac{\lambda_j^2 n_j}{32\sigma_j^2 x_{\max}^2}) + \sum_{i \in [N]} d \exp\left(-\frac{\tilde{\psi} n_i}{8dx_{\max}^2}\right)\right)$, for any $\lambda_j > 0$ and $0 < \eta \leq 1/2 - 1/C_0 - \zeta$ with some constant $C_0 > 2$.

The proof is provided in Appendix A.3. Note that this result is nearly identical to Theorem 1 in the fixed design setting, but it holds with a slightly smaller probability to account for the (unlikely) event that the random design matrices $\{\widehat{\Sigma}^j\}_{j \in [N]}$ are nearly singular.

4.3. Data-Poor Regime

As discussed earlier, multitask learning is especially effective for data-poor tasks, since shared learning can substantially reduce variance in estimation. To illustrate, we consider a task j which has roughly a factor of d^2 fewer observations — i.e., $n_j = \Theta(n_{j'}/d^2)$ for any $j' \neq j$. Note that the data-poor task resides in a *high-dimensional* setting (since $n_j \ll d$), and therefore is unlikely to satisfy Assumption 1 on positive-definiteness. Thus, we replace it with the weaker compatibility condition³, which is standard in the high dimensional literature (Bühlmann and Van De Geer 2011).

DEFINITION 3 (COMPATIBILITY CONDITION). For a constant $\phi > 0$, define the set of matrices

$$\mathcal{C}(\mathcal{S}, \phi) = \{M \in \mathbb{R}^{d \times d} \mid |\mathcal{S}|v^\top Mv \geq \phi \|v_{\mathcal{S}}\|_1^2, \forall \|v_{\mathcal{S}^c}\|_1 \leq 7\|v_{\mathcal{S}}\|_1\}.$$

ASSUMPTION 4 (Compatibility Condition). There exists a constant $\psi > 0$ such that, for the data-poor task j , we have $\widehat{\Sigma}^j \in \mathcal{C}(\bar{\mathcal{I}}_j, \psi)$ for $\bar{\mathcal{I}}_j = \mathcal{I}_{\text{well}} \cup \mathcal{I}_j$ with $\mathcal{I}_j = \{i \in [d] \mid \beta_{(i)}^j \neq \beta_{(i)}^\dagger\}$.

REMARK 4. We also make a minor modification to Algorithm 1 in the data-poor regime: we omit the data-poor task j from the trimmed mean in Step 1 since $\widehat{\beta}_{\text{ind}}^j$ has very high variance. This outcome can also be achieved by increasing the value of the trimming hyperparameter ω .

With this modification, we have the following corollary for a data-poor task j :

COROLLARY 2. Under Assumption 4 for the data-poor task j and Assumption 1 for all other tasks $i \neq j$, the estimator $\widehat{\beta}_{RM}^j$ in Algorithm 1 satisfies

$$\|\widehat{\beta}_{RM}^j - \beta^j\|_1 = \tilde{\mathcal{O}}\left(\frac{s}{\sqrt{n_j}}\right),$$

with at least a probability of $1 - \delta$ for any $\delta \geq \exp\left(-\frac{N-1}{9}\left(\frac{C_0-2}{4C_0}\right)^2 + \log(6d)\right)$, for appropriate choices of hyperparameters ζ , η , and λ provided in Appendix A.4.

³ We show that this condition is weaker than the positive definiteness assumption in Lemma 3 of Appendix A.4.

The proof is provided in Appendix A.4, and follows essentially directly from Theorem 1. For the data-poor task, our estimation error depends only *logarithmically* on the feature dimension d (as opposed to linearly for independent OLS). In other words, the RMEstimator *exponentially* reduces the estimation error (see Table 1), which is especially valuable in data-poor problems.

It is worth noting that the error of task j scales *as if* the parameter β^j is s -sparse. However, our parameters are *not* sparse, i.e., $\|\beta^j\|_0 = d$. Rather, we achieve this scaling as a consequence of multitask learning. When related tasks are data-rich, they provide a good estimate of the shared model β^\dagger , which allows us to substantially reduce the dimensionality of our estimation problem by focusing on learning only the task-specific bias term (which is sparse) rather than β^j (which is dense). This intuition aligns with similar settings considered in Bastani (2021), Xu et al. (2021).

4.4. Network Structure

In practice, one may have a large number of tasks, and may wish to choose a subset within which to perform multitask learning. On one hand, increasing the number of tasks N improves estimation of the shared parameter; on the other hand, an increase in N may imply an increase in the task-specific sparsity parameter s as tasks become increasingly dissimilar.

In Appendix A.5, we assume knowledge of a network that captures the similarity between any pair of tasks; this can be inferred based on observed covariates (e.g., geographic distance between hospitals/stores or socio-economic indices of neighborhoods served) or data from past decision-making problems (see, e.g., the disparity matrix in Crammer et al. 2008). Then, for any given task, we can optimize the “similarity radius” of learning problems from which to transfer knowledge. We consider a simple power scaling for this network, where the effective sparsity parameter varies as $s = \min(N^\alpha, d)$ for some $\alpha \geq 0$. In this setting, we find it is optimal to choose the $N = d^{\frac{1}{\alpha+1}}$ most similar tasks; this results in error bounds (see Corollary 6) that scale with the network density α . Further details and results are provided in Appendix A.5.

5. Robust Multitask Contextual Bandits

We now illustrate the utility of the RMEstimator for online learning via simultaneous multitask contextual bandits. This section describes the modified model setup and assumptions for the bandit setting (§5.1-5.2), overviews our proposed RMBandit algorithm (§5.3), and demonstrates improved total and task-specific regret bounds (§5.4), including in the data-poor regime (§5.5).

5.1. Model

Analogous to the formulation in §2, we consider N distinct linear contextual bandit problems (e.g., hospitals in Example 1 or stores in Example 2). The decision-maker at each instance has access to K potential arms (decisions) with uncertain and context-dependent rewards.

Arrivals. Let T be the overall time horizon across all bandit instances. At each time step t , a new observation arrives at one of the N bandit instances, given by the random variable $Z_t \in [N]$ — i.e., instance j receives an arrival with probability p_j , where $\sum_{i=1}^N p_i = 1$. Thus, Z_t follows a categorical distribution $\mathbf{CG}(\mathbf{p})$ with $\mathbf{p} = [p_1 \cdots p_N]$. In expectation, instance j will have $p_j T$ observations. Again, we consider two regimes: (i) the standard regime where $p_j = \Theta(1/N)$ for all $j \in [N]$, and (ii) the data-poor regime where a single instance j has $p_j = \Theta(p_{j'}/d^2)$ for $j' \neq j$. Each observation has a context $X_t \in \mathbb{R}^d$; if $Z_t = j$, then X_t is drawn i.i.d. from an unknown distribution \mathcal{P}_X^j .

Rewards. The reward for pulling arm k for an observation with context vector X_t at instance j is $Y_t = X_t^\top \beta_k^j + \epsilon_t$. Here, each arm k at instance j is parameterized by an unknown arm parameter $\beta_k^j \in \mathbb{R}^d$, and the corresponding noise ϵ_t is an i.i.d. σ_j -subgaussian random variable given $Z_t = j$. We assume that a given arm satisfies sparse heterogeneity across instances — i.e., there exists $\beta_k^\dagger \in \mathbb{R}^d$ such that $\beta_k^j = \beta_k^\dagger + \delta_k^j$, where δ_k^j is sparse (i.e., $\|\delta_k^j\|_0 \leq s$ for some $s \in \mathbb{N}$) for all $k \in [K], j \in [N]$.

Performance. We want a sequential policy π that learns the arm parameters $\{\beta_k^j\}_{k \in [K], j \in [N]}$ over time, in order to maximize expected reward for each arrival. The overall policy π is composed of sub-policies $\pi_t^j : \mathcal{X}^j \rightarrow [K]$ at each instance j ; we use π_t to represent the arm played at time t . We measure the performance of π by its cumulative expected regret (Lai and Robbins 1985), modified to extend across multiple heterogeneous bandit instances. In particular, when $Z_t = j$ (an observation arrives at instance j), we compare ourselves to the oracle policy π_*^j at instance j , which knows the arm parameters $\{\beta_k^j\}_{k \in [K]}$ in advance. Naturally, π_*^j chooses the arm with the best expected reward, i.e. $\pi_*^j(X_t) = \arg \max_{k \in [K]} X_t^\top \beta_k^j$ for any X_t . The expected regret incurred by pulling arm $\pi_t = k$ at time t given an arrival at instance j is thus

$$r_t^j = \mathbb{E} \left[\max_{k' \in [K]} (X_t^\top \beta_{k'}^j) - X_t^\top \beta_k^j \mid Z_t = j \right],$$

which is simply the difference between the expected reward of using π_*^j and π_t^j . Further taking the expectation over the randomness in where the observation arrives, the expected regret of an overall policy composed of sub-policies $\{\pi_t^j\}_{j \in [N]}$ at time t is

$$r_t = \sum_{j \in [N]} \mathbb{P}[Z_t = j] r_t^j = \sum_{j \in [N]} p_j r_t^j.$$

Our goal is to derive a policy that minimizes the cumulative expected regret $R_T = \sum_{t=1}^T r_t$ across all instances. We also study the instance-specific cumulative regret $R_T^j = \sum_{t=1}^T p_j r_t^j$ for each $j \in [N]$.

5.2. Assumptions

As discussed earlier, we embed our robust multitask estimator into the linear contextual bandit setting studied in Goldenshluger and Zeevi (2013), Bastani and Bayati (2020); therefore, our next four assumptions are directly adapted from this literature.

First, we have a standard assumption that the features and arm parameters are bounded.

ASSUMPTION 5 (Boundedness). *There exist constants $x_{\max} > 0$ and $b > 0$ such that $\|X\|_{\infty} \leq x_{\max}$ for any $X \in \mathcal{X}^j, j \in [N]$ and $\|\beta_k^j\|_1 \leq b$ for any $k \in [K], j \in [N]$.*

Our second assumption is that, for each instance $j \in [N]$, our K arms can be split into two mutually exclusive sets: (i) optimal arms $k \in \mathcal{K}_{\text{opt}}^j$ that are *strictly* optimal in expected reward (by at least h) for any contexts drawn from a set $U_k^j \subset \mathcal{X}^j$ with positive support on \mathcal{P}_X^j , or (ii) suboptimal arms $k \in \mathcal{K}_{\text{sub}}$ that are *strictly* suboptimal in expected reward (by at least h) for all contexts in \mathcal{X}^j . In other words, we assume that every arm is either optimal (by at least h) for at least *some* individuals, or suboptimal for *all* individuals (by at least h). This assumption will ensure that every arm in $\mathcal{K}_{\text{opt}}^j$ will roughly receive at least $p_* p_j T$ samples and quickly learn accurate parameters under a regret-minimizing policy for instance j .

ASSUMPTION 6 (Arm Optimality). *All K arms at any given instance j belong to one of two mutually exclusive sets: $\mathcal{K}_{\text{opt}}^j$ of optimal arms or $\mathcal{K}_{\text{sub}}^j$ of suboptimal arms. There exists some $h > 0$ such that: (i) each $k \in \mathcal{K}_{\text{sub}}^j$ satisfies $X^\top \beta_k^j < \max_{k' \neq k} X^\top \beta_{k'}^j - h$ for any context $X \in \mathcal{X}^j$, and (ii) each $k \in \mathcal{K}_{\text{opt}}^j$ is optimal on a set of contexts*

$$U_k^j = \{X \in \mathcal{X}^j \mid X^\top \beta_k^j > \max_{k' \neq k} X^\top \beta_{k'}^j + h\},$$

which has positive measure, i.e., $\mathbb{P}[X \in U_k^j \mid Z = j] \geq p_$ for some constant $p_* > 0$.*

Our third assumption ensures that linear regression is feasible within the set U_k^j ; this is a mild assumption since it is with respect to the *true* covariance matrix, which only requires that no features are perfectly collinear in this set. In contrast, the *sample* covariance matrix may not be positive-definite at time t , if we have observed too few samples from that instance.

ASSUMPTION 7 (Positive-Definiteness). *For any arm $k \in [K]$ and instance $j \in [N]$, the true covariance matrix $\Sigma_k^j = \mathbb{E}[XX^\top \mid X \in U_k^j, Z = j]$ satisfies $\lambda_{\min}(\Sigma_k^j) \geq \psi$ for some constant $\psi > 0$.*

Our fourth assumption is a margin condition that ensures that the density of the context distribution \mathcal{P}_X^j for each instance j is bounded near a decision boundary (i.e., the hyperplane given by $\{X \in \mathcal{X}^j \mid X^\top \beta_{k'}^j = X^\top \beta_k^j\}$ for any pair $k' \neq k$). It allows for any bounded, continuous features, as well as any discrete features with a finite number of values.

ASSUMPTION 8 (Margin Condition). *For any arms k and k' of any instance $j \in [N]$, there exists a constant $L > 0$ such that $\mathbb{P}[|X^\top(\beta_k^j - \beta_{k'}^j)| \leq \kappa \mid Z = j] \leq L\kappa$ for any $\kappa > 0$.*

REMARK 5. Our regret bounds straightforwardly extend under a more general α -margin condition (see, e.g., Bastani et al. 2021a), as shown in Appendix C.5.

The assumptions thus far are standard and have been adapted directly from the bandit literature. We now introduce a new assumption motivated by our multitask setting. In general, an arm k can be optimal (belong to $\mathcal{K}_{\text{opt}}^j$) at one bandit instance j and be suboptimal (belong to $\mathcal{K}_{\text{sub}}^{j'}$) at a neighboring instance j' . This implies that we will observe $\mathcal{O}(p_j T)$ samples from arm k at instance j but only $\mathcal{O}(\log(p_{j'} T))$ samples at instance i under a regret-minimizing policy; in other words, instance j cannot effectively transfer knowledge from instance i about arm k . Thus, we impose that if an arm $k \in [K]$ is optimal for *any* instance j , it is also optimal for at least *some* subset of the other instances so that we have enough observations to enable multitask learning.

ASSUMPTION 9 (Optimality Density). *The set of bandit instances where arm k is an optimal arm, i.e., $\mathcal{W}_k = \{j \in [N] \mid k \in \mathcal{K}_{\text{opt}}^j\}$, has cardinality of either 0 or at least ρN for some constant $\rho > 0$ for any $k \in [K]$.*

5.3. Algorithm Description

Next, we propose the RMBandit algorithm (Algorithm 2), which leverages the RMEstimator to efficiently learn across N simultaneous linear contextual bandit instances. In this section, we drop the subscript RM and denote our multitask estimator as simply $\hat{\beta}_k^j$ for arm k and instance j .

Following prior work, the RMBandit manages the exploration-exploitation tradeoff using a small amount ($\mathcal{O}(\log(T))$) of forced random exploration in each instance $j \in [N]$. Furthermore, for each instance j and arm $k \in [K]$, it trades off between (i) an unbiased *forced-sample* estimator, which is trained only on forced random samples, and (ii) a potentially biased *all-sample* estimator, which is trained on all observations for arm k . Instead of using LASSO (Bastani and Bayati 2020) or OLS (Goldenshluger and Zeevi 2013) for these estimators, we use our RMEstimator.

This introduces two important challenges. First, our multitask estimator leverages data *across* instances, which induces (previously absent) correlations between our arm parameter estimates $\{\hat{\beta}_k^j\}_{j \in [N]}$ for a fixed arm k . However, our error bound for the trimmed mean estimator (Proposition 4) requires that our estimates across instances be independent in order to recover a reasonable estimate of the shared model β^\dagger . Thus, we introduce a new *batching* strategy, where we only perform parameter updates in batches rather than after every time step. This ensures that our arm parameter estimates in the current batch are independent conditioned on the observations from previous batches. Importantly, this batching strategy does not change the convergence rates (and therefore regret), and has the added advantage of being far more computationally tractable.

Second, our robust multitask estimator requires two hyperparameters: the trimming hyperparameter ω and the LASSO regularization parameter λ (see Algorithm 1). We specify a *trimming path* for ω_t to dynamically trade off bias and variance over time, in order to control the convergence of our robust multitask estimators. Intuitively, we trim less for small t (when we have little

data) to reduce variance at the cost of admitting “small” corruptions; as t increases (when we have collected more data), we trim more aggressively to eliminate even small corruptions that can bias our estimates. For λ_t , the path follows that derived in Bastani and Bayati (2020) correspondingly.

In more detail, we split the time horizon T into disjoint sequential batches $\bigcup_{m \geq 0} \mathcal{B}_m$. The initial batch \mathcal{B}_0 has size $q \log(T)$ for some tuning parameter q , and the following batches iteratively double in length (i.e., the m^{th} batch \mathcal{B}_m has size $|\mathcal{B}_m| = 2^{m-1}|\mathcal{B}_0|$), which yields a total of M batches with

$$M = \left\lceil \log_2 \left(\frac{T}{q \log(T)} \right) \right\rceil. \quad (9)$$

We define $\mathcal{B}_{\bar{m}} = \bigcup_{l=0}^m \mathcal{B}_l$ as the union of all the batches up to batch m . We denote our **RMEstimator** (Algorithm 1) at instance j for arm k as

$$\hat{\beta}_k^j(\mathcal{B}, \lambda, \omega),$$

where the first argument indicates the training sample, i.e., all observations where we pulled arm k in batch \mathcal{B} ; the remaining arguments are hyperparameters, i.e., the LASSO regularization parameter λ and the trimmed mean parameter ω .

Algorithm 2 Robust Multitask Bandit (RMBandit)

Input parameters: $\omega_0, \lambda_0, \zeta_{1,0}, \eta_{1,0}, \lambda_{1,0}, q, h$

Set $M = \left\lceil \log_2 \left(\frac{T}{q \log(T)} \right) \right\rceil$, $\mathcal{B}_0 = [q \log(T)]$, and $\mathcal{B}_m = \{t \in [T] \mid 2^{m-1}|\mathcal{B}_0| < t \leq 2^m|\mathcal{B}_0|\}$ for $m \in [M]$

Initialize $\lambda_{0,j} = \lambda_0 / \sqrt{|\mathcal{B}_0^j|}$

for $t \in [T]$ **do**

 Observe an arrival at instance $j = Z_t \sim \mathbf{CG}(\mathbf{p})$, and the corresponding context $X_t \sim \mathcal{P}_X^j$

if $t \in \mathcal{B}_0$ **then**

 Pull arm $\pi_t = \left((\sum_{r \in [t]} \mathbb{1}(Z_r = j) - 1) \bmod K \right) + 1$

else if $t \in \mathcal{B}_m$ **then**

 Let $\mathcal{K} = \left\{ k \in [K] \mid X_t^\top \hat{\beta}_k^j(\mathcal{B}_0, \lambda_{0,j}, \omega_0) \geq \max_{i \in [K]} X_t^\top \hat{\beta}_i^j(\mathcal{B}_0, \lambda_{0,j}, \omega_0) - h/2 \right\}$

 Pull arm $\pi_t = \arg \max_{k \in \mathcal{K}} X_t^\top \hat{\beta}_k^j(\mathcal{B}_{m-1}, \lambda_{1,j,m-1}, \omega_{1,m-1})$

end if

 Observe reward $Y_t = X_t^\top \beta_{\pi_t}^j + \epsilon_t$

if $t = |\mathcal{B}_{\bar{m}}|$ for $m \in [M]$ (i.e., when \mathcal{B}_m ends) **then**

 Update $\zeta_{1,m} = \zeta_{1,0}$, $\eta_{1,m} = \eta_{1,0} \sqrt{\log(d \min_{i \in [N], |\mathcal{B}_m^i| > 0} |\mathcal{B}_m^i|)}$, and $\omega_{1,m} = \zeta_{1,m} + \eta_{1,m}$

 Update $\lambda_{1,j,\bar{m}} = \lambda_{1,0} \sqrt{\log(d|\mathcal{B}_{\bar{m}}^j|)/|\mathcal{B}_{\bar{m}}^j|}$ for each $j \in [N]$

end if

end for

Strategy. In our initial batch \mathcal{B}_0 , we deterministically forced-sample each arm k of instance j when an individual is observed at instance j (i.e., when $Z_t = j$). At the end of this initial batch, we obtain a forced-sample estimator $\hat{\beta}_k^j(\mathcal{B}_0, \lambda_{0,j}, \omega_0)$ for each $j \in [N]$ and $k \in [K]$; this forced-sample estimator remains fixed for the entire time horizon T . On the other hand, we also maintain an all-sample estimator $\hat{\beta}_k^j(\mathcal{B}_{\bar{m}}, \lambda_{1,j,\bar{m}}, \omega_{1,m})$ for each $j \in [N]$ and $k \in [K]$; this estimator is periodically re-trained with updated hyperparameters at the end of each batch m and thereby fixed for the following batch $m + 1$. One distinction of this estimator from Algorithm 1 is that the trimmed mean estimator in Step 1 is built upon only data from \mathcal{B}_m such that arm parameter estimates are independent. However, in Step 2 we still use all the data of the target instance from $\mathcal{B}_{\bar{m}}$, of which the samples might be correlated across instances.

The algorithm is executed as follows. If $t \in \mathcal{B}_m$ and a new arrival is observed at instance j , we first use the forced-sample estimators to find the highest estimated reward achievable among the K arms at instance j . These estimates allow us to identify a subset of arms $\mathcal{K} \subseteq K$ whose rewards are within $h/2$ of the estimated optimal reward. Then, within this set, we pull the arm $k \in \mathcal{K}$ that has the highest estimated reward according to the all-sample estimators. For a more detailed description of this approach, see Bastani and Bayati (2020).

5.4. Regret Analysis

First, Theorem 2 below bounds the total regret across all N bandit instances.

THEOREM 2. *When $N = \Omega(\log(d)\log(T))$, the total cumulative expected regret of all instances up to time T for RMBandit satisfies*

$$R_T = \mathcal{O}\left(Kd(sN + d)\log(N)\log^2(dT/N)\right),$$

for appropriate choices of hyperparameters $\omega_0, \zeta_{1,0}, \eta_{1,0}, \lambda_0, \lambda_{1,0}$, and q provided in Appendix C.

We provide a detailed proof in Appendix C. Note that we assume that N is not *too* small; our approach is designed for problems where there are at least a few distinct problem instances to enable multitask learning. In §6, we show improved empirical results even for modest values of N .

REMARK 6. The above result is based on our margin condition in Assumption 8. As noted in §5.2, we consider a more general α -margin condition in Appendix C.5, and derive corresponding regret bounds that may scale polynomially in T depending on the strength of the assumption.

Next, we consider the regret of RMBandit for a single instance j . To make a direct comparison to existing regret bounds with time horizon T , we rescale the expected time horizon⁴ for instance j to be T as well, i.e., since we expect $p_j = \Theta(1/N)$ fraction of the total arrivals at instance j , we scale our total horizon as $T/p_j = \Theta(NT)$.

⁴ Note that, given a fixed time horizon across all N instances, the time horizon (i.e., number of observations) for a single instance j is a random variable since the distribution of arrivals across instances $(\{Z_t\}_{t=1}^T)$ is a random process.

COROLLARY 3. *Consider a total time horizon of $T/p_j = \Theta(NT)$, so instance j has expected time horizon T . Then, the cumulative expected regret of instance j for **RMBandit** satisfies*

$$R_T^j = \mathcal{O}\left(Kd(s + d/N)\log(N)\log^2(dT)\right),$$

for the conditions and the hyperparameter choices specified in Theorem 2.

The proof is provided in Appendix C.6. Prior literature shows that such an instance would achieve regret that scales as $\mathcal{O}(d^2 \log^{\frac{3}{2}}(d) \log(T))$ (Bastani and Bayati 2020). In contrast, our upper bound on the regret for instance j using **RMBandit** is smaller by a factor of d , but larger by a factor of $\log(T)$; ⁵ this is a substantial improvement for even a moderate context dimension d . In other words, by appropriately managing the bias-variance tradeoff over time, the **RMBandit** achieves analogous improvements to online learning as we observed with the **RMEstimator** in offline learning.

5.5. Data-Poor Regime

We now turn to the data-poor regime where we expect magnified benefits to multitask learning (matching our offline results in §4.3). Let the target instance j be data-poor, i.e., $p_j = \Theta(p_{j'}/d^2) = \Theta(1/(d^2N))$. Again, to make a direct comparison to existing regret bounds, we rescale our total horizon as $T/p_j = \Theta(d^2NT)$, implying an expected time horizon of T for instance j .

COROLLARY 4. *Consider a total time horizon of $T/p_j = \Theta(d^2NT)$, so the data-poor instance j has expected time horizon T . Then, the cumulative expected regret of instance j for **RMBandit** satisfies*

$$R_T^j = \mathcal{O}\left(Ks^2\log(N)\log^2(dT)\right),$$

for the conditions specified in Theorem 2 and the hyperparameter choices in Appendix C.7.

We provide a proof in Appendix C.7. The above result shows that **RMBandit** attains a regret bound that only scales *logarithmically* in the context dimension d — i.e., we obtain an *exponential* reduction in regret, which is especially valuable in data-poor problems. Note that the regret of instance j scales as if the arm parameters $\{\beta_k^j\}$ are s -sparse (see, e.g., Bastani and Bayati (2020)) while our arm parameters are dense; this intuition aligns with our offline result (Corollary 2).

6. Experiments

We now illustrate the value of our approach on synthetic and real datasets in both offline and online learning settings. For the real experiments, we analyze a patient health risk prediction task in the offline setting; in the online setting, we focus on two well-studied applications of bandits — i.e., patient personalized interventions, and dynamic pricing.

⁵ The extra factor of $\log(T)$ is likely an analytical limitation that arises because the **RMBandit** uses LASSO, e.g., the high-dimensional contextual bandit also attains a regret that scales as $\log^2(T)$ (Bastani and Bayati 2020).

6.1. Multitask Learning

We first analyze the efficiency of our **RMEstimator** in the offline setting. More specifically, we simulate the following linear regression algorithms: (i) without multitask learning: OLS (Hastie et al. 2009), LASSO (Tibshirani 1996a), and (ii) with multitask learning: group LASSO (Argyriou et al. 2008, Lounici et al. 2009), nuclear-norm regularization (Negahban and Wainwright 2011, Pontil and Maurer 2013), pooling (Crammer et al. 2008, Ben-David et al. 2010) and **RMEstimator**. The first two approaches treat each task independently via OLS and LASSO respectively, while the rest share knowledge across N tasks to learn for the target task through group LASSO regularization, nuclear-norm regularization, pooling then OLS, and our **RMEstimator** respectively.

Synthetic. Figure 4 compares the prediction error across three different settings of our formulation in §2 with varying N, d and s . Appendix E.1 provides additional details.

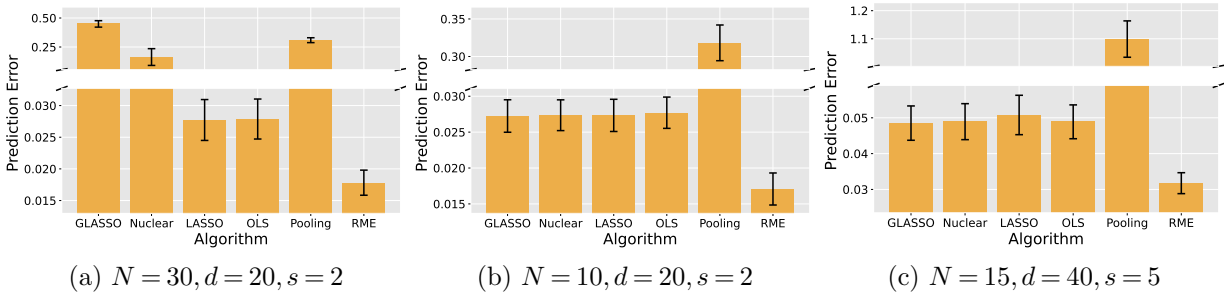


Figure 4 Bars depict prediction error of one task averaged over 20 trials, with corresponding 95% confidence intervals. ‘GLASSO’ is group LASSO, ‘Nuclear’ is nuclear-norm regularization, and ‘RME’ is **RMEstimator**.

We observe that the **RMEstimator** substantially reduces prediction error across the board under varying conditions. OLS/LASSO miss out on the opportunity to perform shared learning. Classical estimators that do perform multitask learning impose structural assumptions that are not met by our synthetic data (and more importantly, appear unwarranted in the real dataset we study in the next subsection) and therefore also perform poorly — e.g., pooling ignores task-specific bias, group LASSO assumes predictive models across tasks are sparse with shared support, and nuclear-norm regularization postulates a shared latent low-rank structure across tasks.

These results are unsurprising since our synthetic data satisfies sparse heterogeneity, so we expect **RMEstimator** to outperform other estimators that do not explicitly leverage this structure. The next experiment examines real electronic medical record data, which may not satisfy our assumptions.

In Appendix F.1, we comprehensively explore the robustness of the **RMEstimator** against a range of model parameters (i.e., N , s , n_i/n_j) and assumptions on the task specific bias terms $\{\delta_j\}_{j \in [N]}$ (their magnitude, alignment of support across tasks, approximate sparsity) through additional

simulations. We find the predictive accuracy of **RMEstimator** improves with large N , small s , large n_i/n_j , and when the support of the task-specific bias terms are either very well- or poorly-aligned.

Risk Prediction in Health Data. Diabetes is a leading cause of severe health complications such as cardiovascular disease, stroke, and chronic kidney disease (Ismail et al. 2021). Thus, there is significant interest in leveraging machine learning for early detection of (Type II) diabetes, in order to improve treatment outcomes (Zhang et al. 2020). However, significant evidence shows that machine learning models trained on one health system can perform poorly on a different health system (Quiñonero-Candela et al. 2008, Subbaswamy and Saria 2020); this can be due to dataset shifts such as changes in patient demographics, disease prevalence, measurement timing, equipment, and treatment patterns. Thus, it is important to train provider-specific risk models.

In this experiment, we use electronic medical record data across $N = 13$ healthcare providers to learn a good diabetes risk prediction model for a single provider. After basic preprocessing, we have approximately 80 patient-specific features constructed from information available before the most recent visit (e.g., past diagnoses, procedures and medications); our outcome is an indicator variable for whether the patient was diagnosed with diabetes during the most recent visit. We aim to learn the best linear classifier, and evaluate different methods based on the diagnosis accuracy over time. Appendix E.2 provides additional details on the setup. We compare the prediction accuracy over two of the hospitals in Figure 7, one with 301 unique patients observed during the sample period and the other with 246 unique patients.

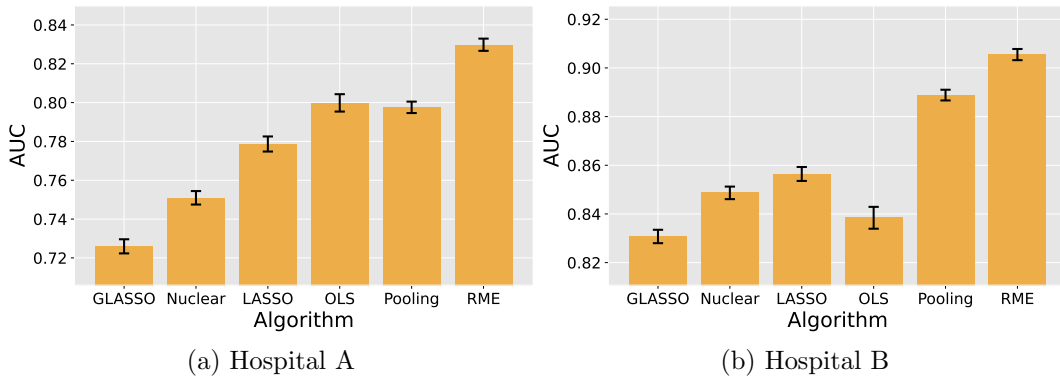


Figure 5 Bars depict fraction of correct diagnoses at one hospital (averaged over 1,000 trials), with 95% confidence intervals. Hospitals A and B have 301 and 246 unique patients respectively. ‘GLASSO’ refers to group LASSO, ‘Nuclear’ nuclear-norm regularization, and ‘RME’ our robust multitask estimator.

Figure 7 shows the fraction of correct diagnoses. Once again, we can find that **RMEstimator** achieves the best performance (AUC), improving over independent OLS by 3.7% and 8% at Hospitals A and B respectively. The pooling algorithm outperforms OLS by leveraging more data but is still not competitive with **RMEstimator** ignoring hospital-specific heterogeneity. Group LASSO

and nuclear-norm regularization perform poorly, suggesting that the assumptions imposed may be unwarranted in our dataset.

6.2. Multitask Bandits

Next, we illustrate the performance of our RMBandit algorithm in the online setting. More specifically, we simulate the following linear contextual bandit algorithms: (i) without multitask learning: OLS Bandit (Goldenshluger and Zeevi 2013), LASSO Bandit (Bastani and Bayati 2020), and (ii) with multitask learning: GOBLin (Cesa-Bianchi et al. 2013), Trace-norm Bandit (Cella et al. 2022), a pooling bandit algorithm, and RMBandit. The first two approaches operate N independent bandit instances via either OLS or LASSO. The GOBLin algorithm is a state-of-the-art multitask bandit algorithm that uses a Laplacian matrix and ridge regression to jointly learn the instances, thereby ℓ_2 -regularizing both the parameters and their pairwise differences. It builds upon the OFUL algorithm (Abbasi-Yadkori et al. 2011), which leverages UCB for linear contextual bandits. Besides, we also implement other multitask bandit algorithms such as Trace-norm Bandit using trace-norm regularization (i.e., nuclear-norm regularization), a pooling bandit algorithm that pools observations and then uses OLS Bandit, and our RMBandit algorithm.

Synthetic. Figure 6 shows the expected cumulative regret over time for a single contextual linear bandit instance under varying N, K, d and s . Appendix E.1 provides additional details.

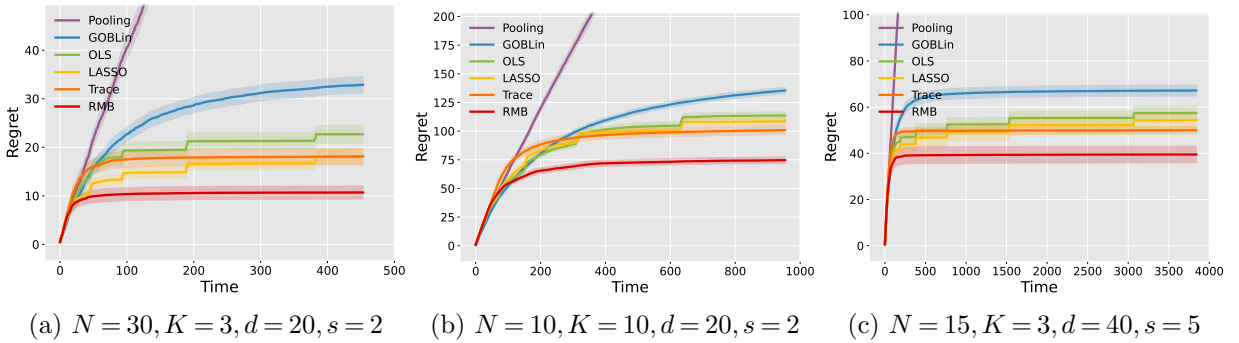


Figure 6 Cumulative regret of a single linear contextual bandit (averaged over 20 trials); shaded regions show 95% confidence intervals. ‘Trace’ refers to Trace-norm Bandit, and ‘RMB’ our RMBandit algorithm.

Similar to the offline setting, we find that RMBandit significantly outperforms other algorithms under sparse heterogeneity. The pooling algorithm ignores task-specific heterogeneity, and the Trace-norm Bandit and GOBLin (which assumes that the arm parameters across instances are close in ℓ_2 norm) impose structure that are not met by our synthetic data. GOBLin performs particularly poorly, possibly because they also build on the UCB algorithm, which is known to over-explore compared to the OLS Bandit (see, e.g., Russo et al. 2017, Bastani et al. 2021a); an interesting future direction is to adapt multitask learning to other bandit algorithms such as Thompson Sampling.

Appendix F.2 explores the robustness of RMBandit against hyperparameter choices ω_0 , $\zeta_{1,0}$, $\eta_{1,0}$, and q . We find that our algorithm’s cumulative regret is robust across specifications, which is important in practice where these hyperparameters might not be well-specified.

The following two experiments examine real datasets, which may not satisfy our assumptions.

Personalized Intervention in Health Data. Using the same medical data as in §6.1, we aim to learn a good diabetes intervention model for each single hospital in an online manner. We consider a simple binary reward that directly evaluates the accuracy of our intervention or classification of patients, i.e., the reward is 1 if the prediction is correct and 0 otherwise; thus, we have two arms, i.e., $K = 2$, in our bandit model that are either to intervene or not intervene on the patient. We aim to learn the best linear classifier online as patient observations accrue, and evaluate different methods based on the classification accuracy over time. We compare the fraction of incorrectly intervened patients over time and show the results of two of the hospitals in Figure 7, one with 355 unique patients observed during the sample period and the other with 176 unique patients. We additionally fit a linear oracle, which leverages all observed data from the provider in hindsight using a leave-one-out approach, representing the best achievable performance within a linear model family. Appendix E.2 provides additional details on the setup.

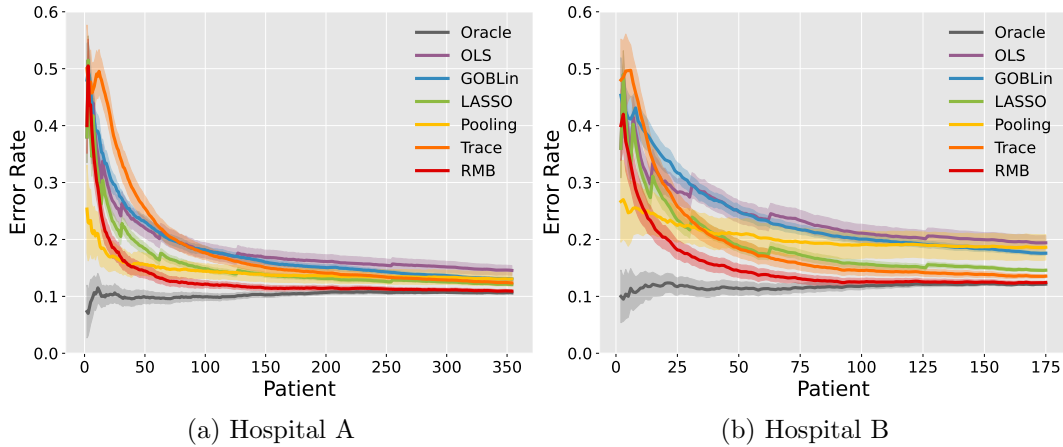


Figure 7 Lines depict the fraction of incorrect interventions averaged over 50 trials of a linear contextual bandit, with shaded regions the corresponding 95% confidence intervals. Hospitals A and B have 355 and 176 unique patients respectively. ‘Trace’ refers to Trace-norm Bandit, and ‘RMB’ our RMBandit algorithm.

We observe that RMBandit performs favorably compared to the other baseline algorithms, and converges to the oracle’s classification accuracy much faster. In particular, our algorithm is insignificantly different from the oracle after observing around 150 and 75 patients respectively, while the other algorithms are all significantly worse than the oracle within the sample period. Notably, the pooling algorithm has good performance on average in the very early stage, likely since our algorithm requires at least some initial data to learn heterogeneous structure across hospitals.

Dynamic Pricing in Retail Data. Contextual bandit algorithms can also naturally be extended to solve dynamic pricing problems with unknown demand (Besbes and Zeevi 2009). We consider such a demand forecasting and price optimization task for food distributors; to this end, we use a publicly available dataset of orders from a meal delivery company.⁶

In this experiment, we use data across $N = 20$ fulfillment centers, serving between six to seven thousand orders each during the sample period. There are 19 features including the intercept, the category and cuisine pertaining to the order, as well as associated promotions. The decision variable is the (continuous) price for the order; rather than arm parameters, there is a single set of unknown parameters (per instance) that aims to predict demand/revenue as a function of price and the observed features. Following the approach of Ban and Keskin (2021), we model the price elasticity of demand as a linear function of the observed features:

$$Y_t = X_t^\top \beta_0^j + p_t \cdot (X_t^\top \beta_1^j) + \epsilon_t.$$

Here, $\{\beta_0^j, \beta_1^j\}$ are the unknown parameters corresponding to instance j ; conditioned on an arrival with context X_t at instance $Z_t = j$, Y_t is the observed revenue for the chosen price p_t and noise ϵ_t . Regret is measured with respect to an oracle that knows $\{\beta_0^j, \beta_1^j\}_{j=1}^N$.

We straightforwardly extend RMBandit and the other baseline bandit algorithms to the dynamic pricing setting using a batched explore-then-commit strategy employed by Ban and Keskin (2021). Figure 8 shows the cumulative regret of RMX (our dynamic pricing analog of RMBandit) compared to other benchmarks including ILQX and ILSX (the LASSO- and OLS-based pricing algorithms introduced in Ban and Keskin 2021), our dynamic pricing analog of GOBLin and Trace-norm Bandit. Appendix E.3 provides further details and pseudocode for our dynamic pricing algorithms.

We observe that RMX performs favorably compared to the dynamic pricing analogs of the other baseline bandit algorithms. Thus, our insights on multitask learning carry over analogously to the dynamic pricing context.

7. Discussion and Conclusions

Decision-makers frequently want to learn heterogeneous treatment effects across many simultaneous experiments. Examples range from learning patient risk across hospitals for personalized interventions (Bastani 2021, Mullainathan and Obermeyer 2017), learning drug effectiveness across combination therapies for clinical trial decisions (Bertsimas et al. 2016), learning COVID-19 risk across travelers for targeting tests (Bastani et al. 2021b), and learning demand across stores for promotion targeting (Baardman et al. 2020, Cohen and Perakis 2018) or dynamic pricing (Bastani

⁶<https://datahack.analyticsvidhya.com/contest/genpact-machine-learning-hackathon-1/>

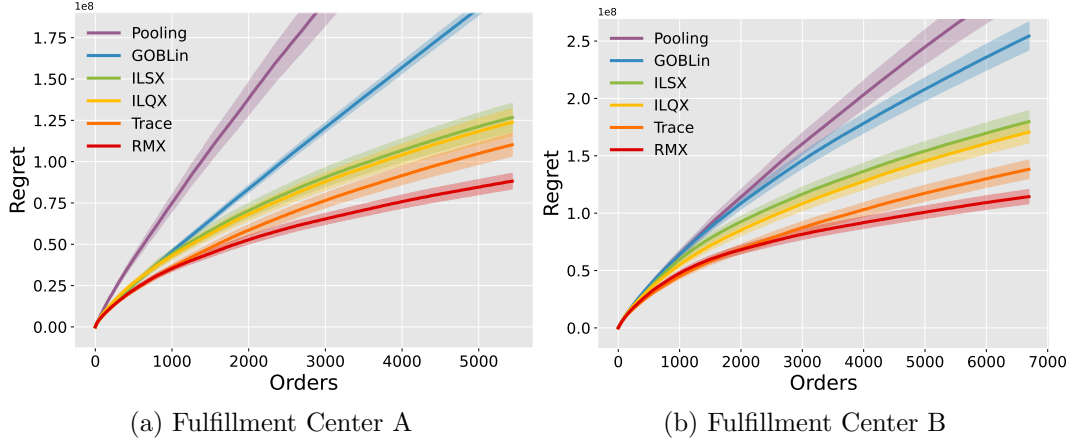


Figure 8 Lines depict the cumulative regret of a linear contextual bandit averaged over 100 trials, with shaded regions the corresponding 95% confidence intervals. ‘Trace’ refers to Trace-norm Bandit, and ‘RMX’ our pricing analog of RMBandit.

et al. 2021c). We propose a novel `RMEstimator` that improves the efficacy of downstream decisions by learning better predictive models with lower sample complexity in the context dimension d . To the best of our knowledge, our work proposes the first combination of robust statistics (to learn across similar instances) and LASSO regression (to debias the results) to yield improved bounds for multitask learning. In the online learning setting, these problems translate to running simultaneous contextual bandit algorithms. To this end, we propose the `RMBandit` algorithm to effectively navigate the exploration-exploitation tradeoff across bandit instances, thereby improving regret bounds in d .

We highlight several features of our proposed approach that make it a particularly attractive solution. First, it is well known that data limitations result in worse model performance, which in turn can imply *unfair* decisions, e.g., in healthcare, such biases disproportionately affect protected groups or minorities due to limited representative data (Rajkomar et al. 2018). A natural approach to alleviating unfairness is to improve the performance of our models for data-poor instances (see, e.g., discussion in Hardt et al. 2016). We show that multitask learning can be especially valuable in such settings — our approach leverages data from data-rich instances to provide an exponential improvement in performance for data-poor instances. Thus, we provide one additional tool (among others) for improving fairness in decision-making.

Second, privacy and regulatory constraints prevent granular data sharing in many applications. A growing literature on *federated learning* studies training statistical models over siloed datasets, while keeping data localized (Li et al. 2020b). While our focus is on multitask learning, our approach satisfies the constraints of federated learning, since we only require sharing aggregate statistics (in this case, OLS regression parameters) across instances. All model training is performed locally at the instance-level and does not require any raw data from other instances.

Third, practical deployment of bandits often precludes real-time updates to the model. For instance, many individuals may appear for service simultaneously (Schwartz et al. 2017) and there may be operational constraints or concerns over model reliability (Bastani et al. 2021b). Our RMBandit algorithm employs a batching strategy that only requires a logarithmic number (in the time horizon T) of model updates, while preserving convergence rates (and therefore regret). Furthermore, it has the added advantage of being far more computationally tractable.

References

- Abbasi-Yadkori, Yasin, Dávid Pál, Csaba Szepesvári. 2011. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems* **24**.
- Alon, Noga, Joel H Spencer. 2004. *The probabilistic method*. John Wiley & Sons.
- Argyriou, Andreas, Theodoros Evgeniou, Massimiliano Pontil. 2008. Convex multi-task feature learning. *Machine learning* **73** 243–272.
- Baardman, Lennart, Setareh Borjian Boroujeni, Tamar Cohen-Hillel, Kiran Panchangam, Georgia Perakis. 2020. Detecting customer trends for optimal promotion targeting. *Manufacturing & Service Operations Management* .
- Ban, Gah-Yi, N Bora Keskin. 2021. Personalized dynamic pricing with machine learning: High-dimensional features and heterogeneous elasticity. *Management Science* **67**(9) 5549–5568.
- Bastani, Hamsa. 2021. Predicting with proxies: Transfer learning in high dimension. *Management Science* **67**(5) 2964–2984.
- Bastani, Hamsa, Mohsen Bayati. 2020. Online decision making with high-dimensional covariates. *Operations Research* **68**(1) 276–294.
- Bastani, Hamsa, Mohsen Bayati, Khashayar Khosravi. 2021a. Mostly exploration-free algorithms for contextual bandits. *Management Science* **67**(3) 1329–1349.
- Bastani, Hamsa, Kimon Drakopoulos, Vishal Gupta, Ioannis Vlachogiannis, Christos Hadjicristodoulou, Pagona Lagiou, Gkikas Magiorkinis, Dimitrios Paraskevis, Sotirios Tsiodras. 2021b. Efficient and targeted covid-19 border testing via reinforcement learning. *Nature* **599**(7883) 108–113.
- Bastani, Hamsa, David Simchi-Levi, Ruihao Zhu. 2021c. Meta dynamic pricing: Transfer learning across experiments. *Management Science* .
- Ben-David, Shai, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning* **79**(1) 151–175.
- Bertsimas, Dimitris, Allison O’Hair, Stephen Relyea, John Silberholz. 2016. An analytics approach to designing combination chemotherapy regimens for cancer. *Management Science* **62**(5) 1511–1531.
- Besbes, Omar, Assaf Zeevi. 2009. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research* **57**(6) 1407–1420.
- Bickel, Peter, Ya’acov Ritov, Alexandre Tsybakov. 2009. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics* 1705–1732.
- Bühlmann, Peter, Sara Van De Geer. 2011. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media.
- Candes, Emmanuel, Terence Tao. 2007. The dantzig selector: statistical estimation when p is much larger than n . *The Annals of Statistics* 2313–2351.
- Caruana, Rich. 1997. Multitask learning. *Machine learning* **28**(1) 41–75.
- Cella, Leonardo, Alessandro Lazaric, Massimiliano Pontil. 2020. Meta-learning with stochastic linear bandits. *International Conference on Machine Learning*. PMLR, 1360–1370.
- Cella, Leonardo, Karim Lounici, Massimiliano Pontil. 2022. Multi-task representation learning with stochastic linear bandits. *arXiv preprint arXiv:2202.10066* .

-
- Cesa-Bianchi, Nicolo, Claudio Gentile, Giovanni Zappella. 2013. A gang of bandits. *arXiv preprint arXiv:1306.0811* .
- Chen, Scott S, David L Donoho, Michael A Saunders. 1995. Atomic decomposition by basis pursuit .
- Cohen, Maxime, Georgia Perakis. 2018. Promotion optimization in retail. *Available at SSRN 3194640* .
- Crammer, Koby, Michael Kearns, Jennifer Wortman. 2008. Learning from multiple sources. *Journal of Machine Learning Research* **9**(8).
- Deshmukh, Aniket Anand, Urun Dogan, Clayton Scott. 2017. Multi-task learning for contextual bandits. *arXiv preprint arXiv:1705.08618* .
- Dobriban, Edgar, Yue Sheng. 2021. Distributed linear regression by averaging. *The Annals of Statistics* **49**(2) 918–943.
- Evgeniou, Theodoros, Charles A Micchelli, Massimiliano Pontil, John Shawe-Taylor. 2005. Learning multiple tasks with kernel methods. *Journal of machine learning research* **6**(4).
- Evgeniou, Theodoros, Massimiliano Pontil. 2004. Regularized multi-task learning. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 109–117.
- Finn, Chelsea, Pieter Abbeel, Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. *International Conference on Machine Learning*. PMLR, 1126–1135.
- Gentile, Claudio, Shuai Li, Giovanni Zappella. 2014. Online clustering of bandits. *International Conference on Machine Learning*. PMLR, 757–765.
- Goldenshluger, Alexander, Assaf Zeevi. 2013. A linear response bandit problem. *Stochastic Systems* **3**(1) 230–261.
- Gupta, Vishal, Nathan Kallus. 2021. Data pooling in stochastic optimization. *Management Science* .
- Hanneke, Steve, Samory Kpotufe. 2020. A no-free-lunch theorem for multitask learning. *arXiv preprint arXiv:2006.15785* .
- Hardt, Moritz, Eric Price, Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems* **29** 3315–3323.
- Hastie, Trevor, Robert Tibshirani, Jerome H Friedman, Jerome H Friedman. 2009. *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer.
- Hsu, Daniel, Sham M Kakade, Tong Zhang. 2011. An analysis of random design linear regression. *arXiv preprint arXiv:1106.2363* .
- Hsu, Daniel, Sham M Kakade, Tong Zhang. 2012. Random design analysis of ridge regression. *Conference on learning theory*. JMLR Workshop and Conference Proceedings, 9–1.
- Ismail, Leila, Huned Materwala, Juma Al Kaabi. 2021. Association of risk factors with type 2 diabetes: A systematic review. *Computational and Structural Biotechnology Journal* **19** 1759–1785.
- Konstantinov, Nikola, Christoph Lampert. 2019. Robust learning from untrusted sources. *International Conference on Machine Learning*. PMLR, 3488–3498.
- Kveton, Branislav, Mikhail Konobeev, Manzil Zaheer, Chih-wei Hsu, Martin Mladenov, Craig Boutilier, Csaba Szepesvari. 2021. Meta-thompson sampling. *arXiv preprint arXiv:2102.06129* .
- Lai, Tze Leung, Herbert Robbins. 1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics* **6**(1) 4–22.
- Li, Jerry. 2019. Lecture 1: Introduction to robustness. <https://jerryzli.github.io/robust-ml-fall19/lec1.pdf>. [Online; accessed 13-April-2021].
- Li, Sai, T Tony Cai, Hongzhe Li. 2020a. Transfer learning for high-dimensional linear regression: Prediction, estimation, and minimax optimality. *arXiv preprint arXiv:2006.10593* .
- Li, Sai, Linjun Zhang, T Tony Cai, Hongzhe Li. 2023. Estimation and inference for high-dimensional generalized linear models with knowledge transfer. *Journal of the American Statistical Association* 1–12.
- Li, Tian, Anit Kumar Sahu, Ameet Talwalkar, Virginia Smith. 2020b. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine* **37**(3) 50–60.

-
- Lounici, Karim, Massimiliano Pontil, Alexandre B Tsybakov, Sara Van De Geer. 2009. Taking advantage of sparsity in multi-task learning. *arXiv preprint arXiv:0903.1468* .
- Lounici, Karim, Massimiliano Pontil, Sara Van De Geer, Alexandre B Tsybakov. 2011. Oracle inequalities and optimal inference under group sparsity. *The annals of statistics* **39**(4) 2164–2204.
- Lugosi, Gabor, Shahar Mendelson. 2021. Robust multivariate mean estimation: the optimality of trimmed mean. *The Annals of Statistics* **49**(1) 393–410.
- Mullainathan, Sendhil, Ziad Obermeyer. 2017. Does machine learning automate moral hazard and error? *American Economic Review* **107**(5) 476–80.
- Negahban, Sahand, Martin J Wainwright. 2011. Estimation of (near) low-rank matrices with noise and high-dimensional scaling .
- Pan, Sinno Jialin, Qiang Yang, et al. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* **22**(10) 1345–1359.
- Pontil, Massimiliano, Andreas Maurer. 2013. Excess risk bounds for multitask learning with trace norm regularization. *Conference on Learning Theory*. PMLR, 55–76.
- Quiñonero-Candela, Joaquin, Masashi Sugiyama, Anton Schwaighofer, Neil D Lawrence. 2008. *Dataset shift in machine learning*. Mit Press.
- Raina, Rajat, Andrew Y Ng, Daphne Koller. 2006. Constructing informative priors using transfer learning. *Proceedings of the 23rd international conference on Machine learning*. ACM, 713–720.
- Rajkomar, Alvin, Michaela Hardt, Michael D Howell, Greg Corrado, Marshall H Chin. 2018. Ensuring fairness in machine learning to advance health equity. *Annals of internal medicine* **169**(12) 866–872.
- Rigollet, Phillippe, Jan-Christian Hütter. 2015. High dimensional statistics. *Lecture notes for course 18S997* **813** 814.
- Rousseeuw, Peter J. 1991. Tutorial to robust statistics. *Journal of chemometrics* **5**(1) 1–20.
- Russo, Daniel, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen. 2017. A tutorial on thompson sampling. *arXiv preprint arXiv:1707.02038* .
- Schwartz, Eric M, Eric T Bradlow, Peter S Fader. 2017. Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science* **36**(4) 500–522.
- Smale, Steve, Ding-Xuan Zhou. 2007. Learning theory estimates via integral operators and their approximations. *Constructive approximation* **26**(2) 153–172.
- Soare, Marta, Ouais Alsharif, Alessandro Lazaric, Joelle Pineau. 2014. Multi-task linear bandits. *NIPS2014 Workshop on Transfer and Multi-task Learning: Theory meets Practice*.
- Subbaswamy, Adarsh, Suchi Saria. 2020. From development to deployment: dataset shift, causality, and shift-stable models in health ai. *Biostatistics* **21**(2) 345–352.
- Tian, Ye, Yang Feng. 2022. Transfer learning under high-dimensional generalized linear models. *Journal of the American Statistical Association* 1–14.
- Tibshirani, Robert. 1996a. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 267–288.
- Tibshirani, Robert. 1996b. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* **58**(1) 267–288.
- Tropp, Joel A. 2015. An introduction to matrix concentration inequalities. *arXiv preprint arXiv:1501.01571* .
- van Herpen, Erica, Erjen van Nierop, Laurens Sloot. 2012. The relationship between in-store marketing and observed sales for organic versus fair trade products. *Marketing Letters* **23**(1) 293–308.
- Wang, Yining, Boxiao Chen, David Simchi-Levi. 2021. Multimodal dynamic pricing. *Management Science* **67**(10) 6136–6152.
- Xu, Kan, Xuanyi Zhao, Hamsa Bastani, Osbert Bastani. 2021. Group-sparse matrix factorization for transfer learning of word embeddings. *International Conference on Machine Learning*. PMLR, 11603–11612.

- Yin, Dong, Yudong Chen, Ramchandran Kannan, Peter Bartlett. 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. *International Conference on Machine Learning*. PMLR, 5650–5659.
- Zhalechian, Mohammad, Esmail Keyvanshokoo, Cong Shi, Mark P Van Oyen. 2022. Online resource allocation with personalized learning. *Operations Research* **70**(4) 2138–2161.
- Zhang, Liying, Yikang Wang, Miaomiao Niu, Chongjian Wang, Zhenfei Wang. 2020. Machine learning for characterizing risk of type 2 diabetes mellitus in a rural chinese population: The henan rural cohort study. *Scientific reports* **10**(1) 1–10.
- Zhang, Tong. 2005. Learning bounds for kernel regression using effective data dimensionality. *Neural Computation* **17**(9) 2077–2098.

Appendix A: Multitask Learning

In this section, we provide proofs for Proposition 4, Theorem 1, Corollary 1 and Corollary 2 in §4.

A.1. Trimmed Mean Estimator

We first prove a tail inequality for our trimmed mean estimator in Proposition 4, which is the key to estimate the global shared parameter β^\dagger .

Proof of Proposition 4 Recall that the indices of the corrupted samples are denoted by $\mathcal{J} \subseteq [N]$ (so the rest are $\mathcal{J}^c = [N] \setminus \mathcal{J}$) with $|\mathcal{J}| < N\zeta$ and $\zeta < 1/2$. By assumption, $\{Z_j\}_{j \in \mathcal{J}^c}$ are independent and σ_j -subgaussian with mean μ respectively. Then, using a Chernoff bound, any uncorrupted sample $j \in \mathcal{J}^c$ satisfies

$$\mathbb{P}[|Z_j - \mu| \geq t] \leq 2 \exp\left(-\frac{t^2}{2\sigma_j^2}\right)$$

for any $t > 0$. Letting $t = \sqrt{2\sigma_j^2 \log(\frac{3}{\eta})}$, it follows that

$$Z_j \notin \left[\mu - \sqrt{2\sigma_j^2 \log(\frac{3}{\eta})}, \mu + \sqrt{2\sigma_j^2 \log(\frac{3}{\eta})}\right]$$

with a probability of at most $2\eta/3$. By Hoeffding's inequality, we have

$$\mathbb{P}\left[\sum_{j \in \mathcal{J}^c} \mathbb{1}(Z_j \notin I) \geq t\right] \leq \exp\left(-\frac{2(t - \sum_{j \in \mathcal{J}^c} p_j)^2}{|\mathcal{J}^c|}\right),$$

where $p_j = \mathbb{P}[Z_j \notin I] \leq 2\eta/3$ and $I = \left[\mu - \max_{j \in \mathcal{J}^c} \sqrt{2\sigma_j^2 \log(\frac{3}{\eta})}, \mu + \max_{j \in \mathcal{J}^c} \sqrt{2\sigma_j^2 \log(\frac{3}{\eta})}\right]$. Taking $t = \eta|\mathcal{J}^c|$, we have

$$\mathbb{P}\left[\sum_{j \in \mathcal{J}^c} \mathbb{1}(Z_j \notin I) \geq \eta|\mathcal{J}^c|\right] \leq \exp\left(-\frac{2\eta^2|\mathcal{J}^c|}{9}\right);$$

in other words, with a high probability, at most η fraction of \mathcal{J}^c are outside a neighborhood I around the mean μ . As a consequence, on the event

$$V = \left\{\sum_{j \in \mathcal{J}^c} \mathbb{1}(Z_j \notin I) \leq \eta|\mathcal{J}^c|\right\},$$

at most $\zeta + \eta$ fraction of the N samples are outside the interval I (recall that ζN samples are corrupted). As a subgaussian distribution might not be symmetric, by trimming the upper and lower ω -quantile of samples with $\omega = \zeta + \eta$, the remaining ones are guaranteed to fall into I .

Suppose the event V holds. Let $\{Z_{j_\iota}\}_{\iota=N\omega+1}^{N(1-\omega)}$ denote the samples after trimming and $\mathcal{T} = \{j_\iota\}_{\iota=N\omega+1}^{N(1-\omega)}$ the corresponding index set (j_ι defined in §3.1). And let $\mathcal{U} = \{j \in [N] \mid Z_j \in I\}$ denote the set of samples that lie in I . Note that $\mathcal{T} \subseteq \mathcal{U}$ on the event of V from our argument above. Then, we have

$$\left|\sum_{j \in \mathcal{T}} (Z_j - \mu)\right| = \left|\sum_{j \in \mathcal{T} \cap \mathcal{U}} (Z_j - \mu)\right| \leq \left|\sum_{j \in \mathcal{T} \cap \mathcal{U} \cap \mathcal{J}^c} (Z_j - \mu)\right| + \left|\sum_{i \in \mathcal{T} \cap \mathcal{U} \cap \mathcal{J}} (Z_j - \mu)\right|. \quad (10)$$

The first term on the RHS of (10) is upper bounded by

$$\left|\sum_{j \in \mathcal{T} \cap \mathcal{U} \cap \mathcal{J}^c} (Z_j - \mu)\right| \leq \left|\sum_{j \in \mathcal{T}^c \cap \mathcal{U} \cap \mathcal{J}^c} (Z_j - \mu)\right| + \left|\sum_{j \in \mathcal{U} \cap \mathcal{J}^c} (Z_j - \mu)\right|.$$

As we remove $2(\zeta + \eta)$ fraction of the samples, we have

$$\left| \sum_{j \in \mathcal{T}^c \cap \mathcal{U} \cap \mathcal{J}^c} (Z_j - \mu) \right| \leq 2(\zeta + \eta)N \max_{j \in \mathcal{J}^c} \sqrt{2\sigma_j^2 \log\left(\frac{3}{\eta}\right)}. \quad (11)$$

Since those samples in \mathcal{J}^c that lie inside the interval I are independent and bounded, Hoeffding's inequality gives

$$\mathbb{P} \left[\left| \frac{1}{|\mathcal{J}^c \cap \mathcal{U}|} \sum_{j \in \mathcal{J}^c \cap \mathcal{U}} (Z_j - \mathbb{E}[Z_j | \mathcal{J}^c \cap \mathcal{U}]) \right| \geq \chi \cdot \max_{j \in \mathcal{J}^c} \sqrt{\sigma_j^2 \log\left(\frac{3}{\eta}\right)} \right] \leq 2 \exp \left(-\frac{|\mathcal{J}^c \cap \mathcal{U}| \chi^2}{4} \right), \quad (12)$$

for any $\chi > 0$. The truncation on these samples introduces a bias of at most

$$|\mathbb{E}[Z_j | \mathcal{J}^c \cap \mathcal{U}] - \mu| = \left| \frac{\mathbb{E}[(Z_j - \mu) \mathbb{1}(Z_j \notin I) | \mathcal{J}^c]}{\mathbb{P}(Z_j \in I | \mathcal{J}^c)} \right| \leq \frac{\mathbb{E}[|Z_j - \mu|^k | \mathcal{J}^c]^{1/k} \mathbb{P}[Z_j \notin I | \mathcal{J}^c]^{1/q}}{\mathbb{P}(Z_j \in I | \mathcal{J}^c)}, \quad (13)$$

where the last inequality uses Hölder's inequality and k, q are such that $1/k + 1/q = 1$. Recall that $\mathbb{P}[Z_j \notin I | \mathcal{J}^c] \leq 2\eta/3$. In addition, $\mathbb{E}[|Z_j - \mu|^k | \mathcal{J}^c]^{1/k} \leq e^{1/e} \sigma_j \sqrt{k}$ for $k \geq 2$ by the property of subgaussian (Rigollet and Hütter 2015). Taking $k = \log(\frac{3}{2\eta})$, inequality (13) gives

$$|\mathbb{E}[Z_j | \mathcal{J}^c \cap \mathcal{U}] - \mu| \leq \frac{8\sigma_j \eta \sqrt{\log(\frac{3}{2\eta})}}{3 - 2\eta}.$$

Then, the high probability bound in (12) implies

$$\mathbb{P} \left[\left| \sum_{j \in \mathcal{J}^c \cap \mathcal{U}} (Z_j - \mu) \right| \geq |\mathcal{J}^c \cap \mathcal{U}| \left(\chi \cdot \max_{j \in \mathcal{J}^c} \sqrt{\sigma_j^2 \log\left(\frac{3}{\eta}\right)} + \max_{j \in \mathcal{J}^c} \frac{8\sigma_j \eta \sqrt{\log(\frac{3}{2\eta})}}{3 - 2\eta} \right) \right] \leq 2 \exp \left(-\frac{|\mathcal{J}^c \cap \mathcal{U}| \chi^2}{4} \right).$$

Further setting $\chi = \eta$ and by our assumption $\eta < 1/2$, we have

$$\mathbb{P} \left[\left| \sum_{j \in \mathcal{J}^c \cap \mathcal{U}} (Z_j - \mu) \right| \geq 5|\mathcal{J}^c \cap \mathcal{U}| \eta \max_{j \in \mathcal{J}^c} \sqrt{\sigma_j^2 \log\left(\frac{3}{\eta}\right)} \right] \leq 2 \exp \left(-\frac{|\mathcal{J}^c \cap \mathcal{U}| \eta^2}{4} \right). \quad (14)$$

Combining (11) and (14), it holds with a high probability that

$$\left| \sum_{j \in \mathcal{T} \cap \mathcal{U} \cap \mathcal{J}^c} (Z_j - \mu) \right| \leq \left(2\sqrt{2}(\zeta + \eta)N + 5\eta|\mathcal{J}^c \cap \mathcal{U}| \right) \max_{j \in \mathcal{J}^c} \sqrt{\sigma_j^2 \log\left(\frac{3}{\eta}\right)}$$

Additionally, the second term on the RHS of (10) has

$$\left| \sum_{j \in \mathcal{T} \cap \mathcal{U} \cap \mathcal{J}} (Z_j - \mu) \right| \leq \zeta N \max_{j \in \mathcal{J}^c} \sqrt{2\sigma_j^2 \log\left(\frac{3}{\eta}\right)}.$$

Thus, we can write

$$\begin{aligned} \left| \frac{1}{|\mathcal{T}|} \sum_{i \in \mathcal{T}} Z_i - \mu \right| &\leq \frac{\max_{j \in \mathcal{J}^c} \sigma_j}{(1 - 2(\zeta + \eta))N} \left(\sqrt{2}(3\zeta + 2\eta)N + 5\eta|\mathcal{J}^c \cap \mathcal{U}| \right) \sqrt{\log\left(\frac{3}{\eta}\right)} \\ &\leq C_0 \max_{j \in \mathcal{J}^c} \sigma_j (3\zeta + 4\eta) \sqrt{\log\left(\frac{3}{\eta}\right)} \end{aligned}$$

with a high probability, where we use $|\mathcal{T}| = (1 - 2(\zeta + \eta))N$, $|\mathcal{J}^c \cap \mathcal{U}| \leq N$ and $\eta \leq 1/2 - 1/C_0 - \zeta$. Since $|\mathcal{J}^c \cap \mathcal{U}| \geq (1 - \eta)|\mathcal{J}^c|$ on the event V , we have

$$\mathbb{P} \left[\left| \frac{1}{|\mathcal{T}|} \sum_{i \in \mathcal{T}} Z_i - \mu \right| \geq C_0 \max_{j \in \mathcal{J}^c} \sigma_j (3\zeta + 4\eta) \sqrt{\log\left(\frac{3}{\eta}\right)} \right] \leq 2 \exp \left(-\frac{N\eta^2}{8} \right),$$

where we use $|\mathcal{J}^c| \geq (1 - \zeta)N$ and $\eta + \zeta < 1/2$. Together with a union bound on the event V , we have

$$\mathbb{P} \left[\left| \frac{1}{|\mathcal{T}|} \sum_{i \in \mathcal{T}} Z_i - \mu \right| \geq C_0 \max_{j \in \mathcal{J}^c} \sigma_j (3\zeta + 4\eta) \sqrt{\log\left(\frac{3}{\eta}\right)} \right] \leq 3 \exp \left(-\frac{N\eta^2}{9} \right). \quad \square$$

A.2. Estimation Error Bound

Then, we provide a proof for Theorem 1 that our **RMEstimator** has small errors with high probability given arbitrary choices of hyperparameters. We begin with the following lemma.

LEMMA 1. *Define the following event for instance j*

$$\mathcal{H}^j = \left\{ \frac{2}{n_j} \|\mathbf{X}^{j\top} \epsilon^j\|_\infty \leq \frac{\lambda_j}{2} \right\}. \quad (15)$$

Then, we have

$$\mathbb{P}[\mathcal{H}^j] \geq 1 - 2d \exp\left(-\frac{\lambda_j^2 n_j}{32\sigma_j^2}\right).$$

Proof of Lemma 1 For any column i of the design matrix \mathbf{X}^j , i.e., $\mathbf{X}_{(\cdot,i)}^j$, we have $\|\frac{1}{\sqrt{n_j}} \mathbf{X}_{(\cdot,i)}^j\|_2 \leq 1$. Then, by Lemma 33, we have

$$\mathbb{P}[(\mathcal{H}^j)^c] = \mathbb{P}\left[\max_{i \in [d]} \frac{1}{\sqrt{n_j}} |\mathbf{X}_{(\cdot,i)}^{j\top} \epsilon^j| \geq \frac{\lambda_j \sqrt{n_j}}{4}\right] \leq d \max_{i \in [d]} \mathbb{P}\left[\frac{1}{\sqrt{n_j}} |\mathbf{X}_{(\cdot,i)}^{j\top} \epsilon^j| \geq \frac{\lambda_j \sqrt{n_j}}{4}\right] \leq 2d \exp\left(-\frac{\lambda_j^2 n_j}{32\sigma_j^2}\right). \quad \square$$

Now, we are ready to prove Theorem 1 by applying Proposition 4 to each dimension of the OLS estimators.

Proof of Theorem 1 First, we show that each OLS estimator $\hat{\beta}_{\text{ind}}^j = (\mathbf{X}^{j\top} \mathbf{X}^j)^{-1} \mathbf{X}^{j\top} Y^j$ constructed in Step 1 is a subgaussian random vector with mean β^j . In particular, the i^{th} component $\hat{\beta}_{\text{ind},(i)}^j$ of $\hat{\beta}_{\text{ind}}^j$ is $\sqrt{\frac{\sigma_j^2}{n_j \psi}}$ -subgaussian since

$$\mathbb{E}[\exp(\lambda(\hat{\beta}_{\text{ind},(i)}^j - \beta_{(i)}^j))] = \mathbb{E}[\exp(\lambda(\mathbf{X}^{j\top} \mathbf{X}^j)^{-1}_{(i,\cdot)} \mathbf{X}^{j\top} \epsilon^j)] \leq \exp\left(\frac{\lambda^2 \sigma_j^2 \|(\mathbf{X}^{j\top} \mathbf{X}^j)^{-1}_{(i,\cdot)}\|_2^2}{2}\right) \leq \exp\left(\frac{\lambda^2 \sigma_j^2}{2n_j \psi}\right),$$

where the last inequality follows since

$$\|(\mathbf{X}^{j\top} \mathbf{X}^j)^{-1}_{(i,\cdot)} \mathbf{X}^{j\top}\|_2^2 = (\mathbf{X}^{j\top} \mathbf{X}^j)^{-1}_{(i,i)} \leq \lambda_{\max}((\mathbf{X}^{j\top} \mathbf{X}^j)^{-1}) = \frac{1}{n_j \lambda_{\min}(\hat{\Sigma}^j)} \leq \frac{1}{n_j \psi}.$$

Now, consider our robust multitask estimator $\{\hat{\beta}_{\text{RM}}^j\}_{j \in [N]}$ computed by Algorithm 1. Recall that for any poorly-aligned component $i \in \mathcal{I}_{\text{poor}}$, the corresponding corrupted subset of instances is $\mathcal{J}_i = \{j \in [N] \mid \beta_{(i)}^j \neq \beta_{(i)}^\dagger\}$. By definition of $\mathcal{I}_{\text{poor}}$, we have $|\mathcal{J}_i| < \zeta N < N/2$. As the data from different instances are mutually independent, the vectors $\{\hat{\beta}_{\text{ind}}^j\}_{j \in [N]}$ are independent. Thus, we can apply Proposition 4 to the trimmed mean of $\{\hat{\beta}_{\text{ind}}^j\}_{j \in [N]}$, where we use the fact that $\hat{\beta}_{\text{ind}}^j$ is $\sqrt{\frac{\sigma_j^2}{n_j \psi}}$ -subgaussian:

$$\mathbb{P}\left[\left|\hat{\beta}_{\text{RM},(i)}^\dagger - \beta_{(i)}^\dagger\right| \geq C_0 (3\zeta + 4\eta) \max_{i \in [N]} \sqrt{\frac{\sigma_i^2}{n_i \psi} \log\left(\frac{3}{\eta}\right)}\right] \leq 3 \exp\left(-\frac{N\eta^2}{9}\right).$$

By a union bound over $i \in \mathcal{I}_{\text{poor}}$, we have

$$\mathbb{P}\left[\left\|\left(\hat{\beta}_{\text{RM}}^\dagger - \beta^\dagger\right)_{\mathcal{I}_{\text{poor}}}\right\|_1 \geq C_0 d (3\zeta + 4\eta) \max_{i \in [N]} \sqrt{\frac{\sigma_i^2}{n_i \psi} \log\left(\frac{3}{\eta}\right)}\right] \leq 3d \exp\left(-\frac{N\eta^2}{9}\right), \quad (16)$$

where $\beta_{\mathcal{I}}$ given an index set \mathcal{I} is defined at the beginning of §2. Next, note that

$$Y^j = \mathbf{X}^j (\beta^\dagger + \delta^j) + \epsilon^j = \mathbf{X}^j \left((\beta_{\mathcal{I}_{\text{poor}}}^\dagger + \hat{\beta}_{\text{RM}, \mathcal{I}_{\text{well}}}^\dagger) + (\beta_{\mathcal{I}_{\text{well}}}^\dagger - \hat{\beta}_{\text{RM}, \mathcal{I}_{\text{well}}}^\dagger + \delta^j) \right) + \epsilon^j,$$

where $\beta_{\mathcal{I}_{\text{well}}}^\dagger - \hat{\beta}_{\text{RM}, \mathcal{I}_{\text{well}}}^\dagger + \delta^j$ is $((1/\zeta + 1)s)$ -sparse — in particular, letting

$$\bar{\mathcal{I}}_j = \mathcal{I}_{\text{well}} \cup \mathcal{I}_j, \quad (17)$$

where $\mathcal{I}_j = \{i \in [d] \mid \beta_{(i)}^j \neq \beta_{(i)}^\dagger\}$ are the components of β^j that do not equal β^\dagger , then we have $|\bar{\mathcal{I}}_j| \leq (1/\zeta + 1)s$. Besides, $\beta_{\mathcal{I}_{\text{poor}}}^\dagger + \hat{\beta}_{\text{RM}, \mathcal{I}_{\text{well}}}^\dagger$ is closely approximated by $\hat{\beta}_{\text{RM}}^\dagger$ as shown in (16). Therefore, after computing $\hat{\beta}_{\text{RM}}^\dagger$ in Step 1 in §3.2, we use LASSO to recover the sparse vector $\beta_{\mathcal{I}_{\text{well}}}^\dagger - \hat{\beta}_{\text{RM}, \mathcal{I}_{\text{well}}}^\dagger + \delta^j$ to efficiently estimate β^j .

The following steps are adapted from the LASSO analysis in Chapter 6.2 of Bühlmann and Van De Geer (2011). The basic inequality of LASSO in Step 2 of our algorithm is

$$\frac{1}{n_j} \|\mathbf{X}^j \hat{\beta}_{\text{RM}}^j - Y^j\|_2^2 + \lambda_j \|\hat{\beta}_{\text{RM}}^j - \hat{\beta}_{\text{RM}}^\dagger\|_1 \leq \frac{1}{n_j} \|\mathbf{X}^j \beta^j - Y^j\|_2^2 + \lambda_j \|\beta^j - \hat{\beta}_{\text{RM}}^\dagger\|_1.$$

Plugging in $Y^j = \mathbf{X}^j \beta^j + \epsilon^j$, and conditioned on \mathcal{H}^j in (15), we have

$$\begin{aligned} \frac{1}{n_j} \|\mathbf{X}^j (\hat{\beta}_{\text{RM}}^j - \beta^j)\|_2^2 + \lambda_j \|\hat{\beta}_{\text{RM}}^j - \hat{\beta}_{\text{RM}}^\dagger\|_1 &\leq \frac{2}{n_j} \epsilon^{j\top} \mathbf{X}^j (\hat{\beta}_{\text{RM}}^j - \beta^j) + \lambda_j \|\beta^j - \hat{\beta}_{\text{RM}}^\dagger\|_1 \\ &\leq \frac{2}{n_j} \|\mathbf{X}^{j\top} \epsilon^j\|_\infty \|\hat{\beta}_{\text{RM}}^j - \beta^j\|_1 + \lambda_j \|\beta^j - \hat{\beta}_{\text{RM}}^\dagger\|_1 \\ &\leq \frac{\lambda_j}{2} \|\hat{\beta}_{\text{RM}}^j - \beta^j\|_1 + \lambda_j \|\beta^j - \hat{\beta}_{\text{RM}}^\dagger\|_1. \end{aligned}$$

Decomposing terms based on $\bar{\mathcal{I}}_j$ and $\bar{\mathcal{I}}_j^c$ and rearranging the inequality, we have

$$\frac{1}{n_j} \|\mathbf{X}^j (\hat{\beta}_{\text{RM}}^j - \beta^j)\|_2^2 + \lambda_j \|(\hat{\beta}_{\text{RM}}^j - \hat{\beta}_{\text{RM}}^\dagger)_{\bar{\mathcal{I}}_j^c}\|_1 \leq \frac{3\lambda_j}{2} \|(\hat{\beta}_{\text{RM}}^j - \beta^j)_{\bar{\mathcal{I}}_j}\|_1 + \frac{\lambda_j}{2} \|(\hat{\beta}_{\text{RM}}^j - \beta^j)_{\bar{\mathcal{I}}_j}\|_1 + \lambda_j \|(\beta^j - \hat{\beta}_{\text{RM}}^\dagger)_{\bar{\mathcal{I}}_j^c}\|_1,$$

so it follows that

$$\frac{1}{n_j} \|\mathbf{X}^j (\hat{\beta}_{\text{RM}}^j - \beta^j)\|_2^2 + \frac{\lambda_j}{2} \|(\hat{\beta}_{\text{RM}}^j - \beta^j)_{\bar{\mathcal{I}}_j^c}\|_1 \leq \frac{3\lambda_j}{2} \|(\hat{\beta}_{\text{RM}}^j - \beta^j)_{\bar{\mathcal{I}}_j}\|_1 + 2\lambda_j \|(\hat{\beta}_{\text{RM}}^\dagger - \beta^j)_{\bar{\mathcal{I}}_j^c}\|_1. \quad (18)$$

Then, adding $\frac{\lambda_j}{2} \|(\hat{\beta}_{\text{RM}}^j - \beta^j)_{\bar{\mathcal{I}}_j}\|_1$ on both sides, we get

$$\frac{1}{n_j} \|\mathbf{X}^j (\hat{\beta}_{\text{RM}}^j - \beta^j)\|_2^2 + \frac{\lambda_j}{2} \|\hat{\beta}_{\text{RM}}^j - \beta^j\|_1 \leq 2\lambda_j \|(\hat{\beta}_{\text{RM}}^j - \beta^j)_{\bar{\mathcal{I}}_j}\|_1 + 2\lambda_j \|(\hat{\beta}_{\text{RM}}^\dagger - \beta^j)_{\mathcal{I}_{\text{poor}}}\|_1, \quad (19)$$

where we use $(\beta_{\mathcal{I}_{\text{well}}}^\dagger - \hat{\beta}_{\text{RM}, \mathcal{I}_{\text{well}}}^\dagger + \delta^j)_{\bar{\mathcal{I}}_j^c} = \mathbf{0}$. By Assumption 1, we have

$$\|(\hat{\beta}_{\text{RM}}^j - \beta^j)_{\bar{\mathcal{I}}_j}\|_1 \leq \sqrt{(1/\zeta + 1)s} \|\hat{\beta}_{\text{RM}}^j - \beta^j\|_2 \leq \sqrt{\frac{(1/\zeta + 1)s}{\psi} \frac{1}{n_j} \|\mathbf{X}^j (\hat{\beta}_{\text{RM}}^j - \beta^j)\|_2^2}.$$

Therefore, we derive from inequality (19) that

$$\frac{1}{2n_j} \|\mathbf{X}^j (\hat{\beta}_{\text{RM}}^j - \beta^j)\|_2^2 + \frac{\lambda_j}{2} \|\hat{\beta}_{\text{RM}}^j - \beta^j\|_1 \leq \frac{2(1/\zeta + 1)s\lambda_j^2}{\psi} + 2\lambda_j \|(\hat{\beta}_{\text{RM}}^\dagger - \beta^j)_{\mathcal{I}_{\text{poor}}}\|_1,$$

where we use the fact that $2ab \leq a^2 + b^2$. Since $\zeta < 1/2$, we further get

$$\|\hat{\beta}_{\text{RM}}^j - \beta^j\|_1 \leq \frac{6s\lambda_j}{\zeta\psi} + 4\|(\hat{\beta}_{\text{RM}}^\dagger - \beta^j)_{\mathcal{I}_{\text{poor}}}\|_1.$$

Combining the above with inequality (16) and Lemma 1, we obtain our result. \square

A.3. Standard Regime

We list the hyperparameter choices for our **RMEstimator** in the standard case as follows. In particular, we take

$$\lambda = \max_{i \in [N]} \sqrt{32\sigma_i^2 \log(\frac{4d}{\delta})}, \quad \zeta = \frac{C_0 - 2}{4C_0} \sqrt{\frac{s}{d}}, \quad \eta = \sqrt{\frac{9 \log(\frac{6d}{\delta})}{N}}.$$

In addition, we take $\lambda_j = \lambda / \sqrt{n_j}$ as in Algorithm 1.

The proofs follow directly from Theorem 1, with additional details listed as follows.

Proof of Corollary 1 Since $\delta \leq 1$ and $d \geq 1$, we have

$$\sqrt{\log(\frac{3}{\eta})} = \sqrt{\frac{1}{2} \log\left(\frac{N}{\log(\frac{6d}{\delta})}\right)} \leq \sqrt{\log(N)}.$$

Therefore, it holds that

$$\begin{aligned} \|\hat{\beta}_{\text{RM}}^j - \beta^j\|_1 &\leq \frac{6\lambda_j s}{\zeta\psi} + C_0 d (3\zeta + 4\eta) \max_{i \in [N]} \sqrt{\frac{\sigma_i^2}{n_i \psi} \log(\frac{3}{\eta})} \\ &\leq \frac{24s}{\zeta\psi} \sqrt{\frac{2\sigma_j^2 \log(\frac{4d}{\delta})}{n_j}} + 3C_0 \zeta d \max_{i \in [N]} \sqrt{\frac{\sigma_i^2 \log(N)}{n_i \psi}} + 12C_0 d \max_{i \in [N]} \sqrt{\frac{\sigma_i^2 \log(N) \log(\frac{6d}{\delta})}{N n_i \psi}}, \end{aligned}$$

with probability at least $1 - \delta$. As n_i 's are similar in scale, it suffices to take $\zeta = \frac{C_0 - 2}{4C_0} \sqrt{\frac{s}{d}}$ to minimize the RHS. Thus, we have with high probability

$$\|\hat{\beta}_{\text{RM}}^j - \beta^j\|_1 \leq \frac{96C_0}{(C_0 - 2)\psi} \sqrt{\frac{2\sigma_j^2 s d \log(\frac{4d}{\delta})}{n_j}} + \frac{3C_0}{4} \max_{i \in [N]} \sqrt{\frac{\sigma_i^2 s d \log(N)}{\psi n_i}} + 12C_0 d \max_{i \in [N]} \sqrt{\frac{\sigma_i^2 \log(N) \log(\frac{6d}{\delta})}{\psi N n_i}}.$$

This holds for any $\delta \geq \exp\left(-\frac{N}{9} \left(\frac{C_0 - 2}{2C_0} \left(1 - \frac{1}{2} \sqrt{\frac{s}{d}}\right)\right)^2 + \log(6d)\right)$, as we require $\eta \leq 1/2 - 1/C_0 - \zeta$. \square

For our result in Proposition 5 for a random design, its proof straightforwardly follows by combining Theorem 1 and the following lemma. In this lemma, we claim that Assumption 1 holds with high probability under Assumption 2 and 3.

LEMMA 2. *We have*

$$\mathbb{P}\left[\lambda_{\min}(\hat{\Sigma}^j) \geq \frac{\tilde{\psi}}{2}\right] \geq 1 - d \exp\left(-\frac{\tilde{\psi} n_j}{8 d x_{\max}^2}\right).$$

The proof follows directly Lemma 34 by noting that $\lambda_{\max}(X_t X_t^\top) \leq \|X_t\|_2^2 \leq d x_{\max}^2$ and setting $t = 1/2$ and $L = d x_{\max}^2$ in Lemma 34.

A.4. Data-Poor Regime

We list the hyperparameter choices in Corollary 2 for our **RMEstimator** in the data-poor case as follows. In particular, we take

$$\lambda = \max_{i \neq j} \sqrt{32\sigma_i^2 \log(\frac{4d}{\delta})}, \quad \zeta = \frac{C_0 - 2}{4C_0}, \quad \eta = \sqrt{\frac{9 \log(\frac{6d}{\delta})}{N - 1}}.$$

In addition, take $\lambda_j = \lambda / \sqrt{n_j}$, as in Algorithm 1.

The following lemma shows that the compatibility condition is weaker than imposing positive-definiteness.

LEMMA 3. *For any matrix $\Sigma \in \mathbb{R}^{d \times d}$ with $\lambda_{\min}(\Sigma) > \phi$, we have $\Sigma \in \mathcal{C}(\mathcal{S}, \phi)$ for any set $\mathcal{S} \subset [d]$.*

Proof of Lemma 3 For any $v \in \mathbb{R}^d$ and $\mathcal{S} \subset [d]$, we have $\|v_{\mathcal{S}}\|_1 \leq \sqrt{|\mathcal{S}|} \|v_{\mathcal{S}}\|_2$. Therefore,

$$|\mathcal{S}| v^T \Sigma v \geq |\mathcal{S}| \phi \|v\|_2^2 \geq |\mathcal{S}| \phi \|v_{\mathcal{S}}\|_2^2 \geq \phi \|v_{\mathcal{S}}\|_1^2. \quad \square$$

Now we are ready to show that Theorem 1 still holds in a data-poor regime under Assumption 4. Note that now in the first stage we calculate our trimmed mean leaving out the data-poor instance j (see Remark 4).

PROPOSITION 6. *Given $\zeta \in (0, 1/2)$, the estimator $\hat{\beta}_{RM}^j$ of data-poor instance j defined in §3.2 satisfies*

$$\|\hat{\beta}_{RM}^j - \beta^j\|_1 \leq \frac{6\lambda_j s}{\zeta\psi} + C_0 d (3\zeta + 4\eta) \max_{i \neq j} \sqrt{\frac{\sigma_i^2}{n_i \psi} \log\left(\frac{3}{\eta}\right)}$$

with at least a probability of $1 - \left(3d \exp\left(-\frac{N\eta^2}{9}\right) + 2d \exp\left(-\frac{\lambda_j^2 n_j}{32\sigma_j^2}\right)\right)$, for any $\lambda_j > 0$ and $0 < \eta \leq 1/2 - 1/C_0 - \zeta$ with some constant $C_0 > 2$.

Proof of Proposition 6 The proof mainly follows that of Theorem 1. In the following, we list the details that differ from Theorem 1.

Now, we apply Proposition 4 to the trimmed mean of $\{\hat{\beta}_{\text{ind}}^\ell\}_{\ell \neq j}$ and obtain

$$\mathbb{P} \left[\left\| (\hat{\beta}_{RM}^\dagger - \beta^\dagger)_{\mathcal{I}_{\text{poor}}} \right\|_1 \geq C_0 d (3\zeta + 4\eta) \max_{i \neq j} \sqrt{\frac{\sigma_i^2}{n_i \psi} \log\left(\frac{3}{\eta}\right)} \right] \leq 3d \exp\left(-\frac{N\eta^2}{9}\right). \quad (20)$$

Following the same steps of Theorem 1 until (18), we have

$$\frac{1}{n_j} \|\mathbf{X}^j (\hat{\beta}_{RM}^j - \beta^j)\|_2^2 + \frac{\lambda_j}{2} \|(\hat{\beta}_{RM}^j - \beta^j)_{\bar{\mathcal{I}}_j^c}\|_1 \leq \frac{3\lambda_j}{2} \|(\hat{\beta}_{RM}^j - \beta^j)_{\bar{\mathcal{I}}_j}\|_1 + 2\lambda_j \|(\hat{\beta}_{RM}^\dagger - \beta^\dagger)_{\mathcal{I}_{\text{poor}}}\|_1. \quad (21)$$

Next, consider the following two cases respectively: (i) $\|(\hat{\beta}_{RM}^j - \beta^j)_{\bar{\mathcal{I}}_j}\|_1 \leq \|(\hat{\beta}_{RM}^\dagger - \beta^\dagger)_{\mathcal{I}_{\text{poor}}}\|_1$, and (ii) $\|(\hat{\beta}_{RM}^j - \beta^j)_{\bar{\mathcal{I}}_j}\|_1 > \|(\hat{\beta}_{RM}^\dagger - \beta^\dagger)_{\mathcal{I}_{\text{poor}}}\|_1$. In the first case, we can obtain from inequality (21) that

$$\|\hat{\beta}_{RM}^j - \beta^j\|_1 \leq 8 \|(\hat{\beta}_{RM}^\dagger - \beta^\dagger)_{\mathcal{I}_{\text{poor}}}\|_1.$$

In the second case, note that we have $\|(\hat{\beta}_{RM}^j - \beta^j)_{\bar{\mathcal{I}}_j^c}\|_1 \leq 7 \|(\hat{\beta}_{RM}^j - \beta^j)_{\bar{\mathcal{I}}_j}\|_1$; thus, given Assumption 4 and $|\bar{\mathcal{I}}_j| \leq (1/\zeta + 1)s$, we have

$$\|(\hat{\beta}_{RM}^j - \beta^j)_{\bar{\mathcal{I}}_j}\|_1 \leq \sqrt{\frac{(1/\zeta + 1)s}{\psi} \frac{1}{n_j} \|\mathbf{X}^j (\hat{\beta}_{RM}^j - \beta^j)\|_2^2}.$$

Then, we can derive from inequality (21) that

$$\frac{\psi}{(1/\zeta + 1)s} \|(\hat{\beta}_{RM}^j - \beta^j)_{\bar{\mathcal{I}}_j}\|_1^2 \leq \frac{1}{n_j} \|\mathbf{X}^j (\hat{\beta}_{RM}^j - \beta^j)\|_2^2 \leq \frac{3\lambda_j}{2} \|(\hat{\beta}_{RM}^j - \beta^j)_{\bar{\mathcal{I}}_j}\|_1,$$

that is,

$$\|(\hat{\beta}_{RM}^j - \beta^j)_{\bar{\mathcal{I}}_j}\|_1 \leq \frac{(1/\zeta + 1)s}{\psi} \frac{3\lambda_j}{2} \leq \frac{9s\lambda_j}{4\zeta\psi},$$

where the last inequality uses $\zeta < 1/2$. Therefore, we have

$$\|(\hat{\beta}_{RM}^j - \beta^j)_{\bar{\mathcal{I}}_j}\|_1 \leq \frac{9s\lambda_j}{4\zeta\psi} + 8 \|(\hat{\beta}_{RM}^\dagger - \beta^\dagger)_{\mathcal{I}_{\text{poor}}}\|_1.$$

Combined the above with (20), our result follows. \square

Next, we prove Corollary 2 under our compatibility condition.

Proof of Corollary 2 Given our choice of the hyperparameters, we derive from Proposition 6 that

$$\|\hat{\beta}_{\text{RM}}^j - \beta^j\|_1 \leq \frac{24s}{\zeta\psi} \sqrt{\frac{2\sigma_j^2 \log(\frac{4d}{\delta})}{n_j}} + 3C_0\zeta \max_{i \neq j} \sqrt{\frac{d^2\sigma_i^2 \log(N)}{n_i\psi}} + 12C_0 \max_{i \neq j} \sqrt{\frac{d^2\sigma_i^2 \log(N) \log(\frac{6d}{\delta})}{Nn_i\psi}},$$

with probability at least $1 - \delta$. Since n_j is assumed to be similar to n_i/d^2 for any $i \neq j$ in magnitude, choosing ζ to be any constant smaller than $1/2 - 1/C_0$ suffices to minimize the RHS and thereby we take $\zeta = \frac{C_0-2}{4C_0}$. Thus, with probability at least $1 - \delta$, we have

$$\|\hat{\beta}^j - \beta^j\|_1 \leq \frac{96C_0}{(C_0-2)\psi} \sqrt{\frac{2\sigma_j^2 s^2 \log(\frac{4d}{\delta})}{n_j}} + \frac{3C_0}{4} \max_{i \neq j} \sqrt{\frac{d^2\sigma_i^2 \log(N)}{\psi n_i}} + 12C_0 \max_{i \neq j} \sqrt{\frac{d^2\sigma_i^2 \log(N) \log(\frac{6d}{\delta})}{Nn_i\psi}}.$$

This holds for any $\delta \geq \exp\left(-\frac{N-1}{9}\left(\frac{C_0-2}{4C_0}\right)^2 + \log(6d)\right)$, as we require $\eta \leq 1/2 - 1/C_0 - \zeta$. \square

For random design, we first show that the compatibility condition in Assumption 4 also holds with high probability under Assumption 2, a result comparable to Lemma 2.

LEMMA 4. *When $n_j \geq 3\log(d)/C'^2$, we have*

$$\mathbb{P}\left[\hat{\Sigma}^j \in \mathcal{C}(\bar{\mathcal{I}}_j, \frac{\tilde{\psi}}{2})\right] \geq 1 - \exp(-C'^2 n_j),$$

where $C' = \min\left\{\frac{1}{2}, \frac{\zeta\tilde{\psi}}{768sx_{\max}^2}\right\}$.

Proof of Lemma 4 First, by Lemma 3 we can imply that $\Sigma^j \in \mathcal{C}(\bar{\mathcal{I}}_j, \tilde{\psi})$ under Assumption 2. Then, the result follows Lemma EC.6 in Bastani and Bayati (2020). \square

Next, we introduce a variant of Proposition 5 in the data-poor regime. The proof straightforwardly combines Lemma 4 and Proposition 5.

COROLLARY 5. *When $n_j \geq 3\log(d)/C'^2$ and given $\zeta \in (0, 1/2)$, the estimator $\hat{\beta}_{\text{RM}}^j$ of data-poor instance j in Algorithm 1 satisfies*

$$\|\hat{\beta}_{\text{RM}}^j - \beta^j\|_1 \leq \frac{6\lambda_j s}{\zeta\phi} + C_0 d (3\zeta + 4\eta) \max_{i \neq j} \sqrt{\frac{\sigma_i^2}{n_i\phi} \log\left(\frac{3}{\eta}\right)}$$

with probability at least $1 - \left(3d \exp(-\frac{N\eta^2}{9}) + 2d \exp(-\frac{\lambda_j^2 n_j}{32\sigma_j^2 x_{\max}^2}) + \sum_{i \neq j} \mathbb{P}\left[\lambda_{\min}(\hat{\Sigma}^i) \leq \phi\right] + \mathbb{P}\left[\hat{\Sigma}^j \in \mathcal{C}(\bar{\mathcal{I}}_j, \tilde{\phi})\right]\right)$, for any $\phi, \tilde{\phi} > 0$, $\lambda_j > 0$ and $0 < \eta \leq 1/2 - 1/C_0 - \zeta$ with some constant $C_0 > 2$.

A.5. Network Structure

We also analyze the impact of an underlying network structure on the estimation error. Specifically, we assume knowledge of a network that captures the similarity between any pair of instances; this can be inferred based on observed covariates (e.g., geographic distance between hospitals/stores or socio-economic indices of neighborhoods served) or data from past decision-making problems (see, e.g., the disparity matrix in Crammer et al. 2008). Then, for any given problem instance, we can optimize the “similarity radius” of learning problems from which to transfer knowledge, resulting in error bounds that scale with the underlying network density.

More specifically, we examine error bound as a function of the underlying network structure, where vertices represent instances and edges capture their pairwise similarity. This sheds light on choosing the number

of instances N for estimation, minimizing a bias-variance tradeoff. In other words, as we incorporate more instances, we reduce variance (since we have more data) but we increase bias (since we are incorporating observations from more disparate sources).

Formally, consider the dependence of the parameter estimation error on the underlying network structure of instances, when available. In particular, we consider a fully-connected network with N vertices (each representing a bandit instance) and edge weights $s_{i,j}$ capturing the pairwise similarities between any two instances $(i, j) \in [N] \times [N]$ as measured by our sparse difference metric, i.e., $\|\beta^j - \beta^i\|_0 \leq s_{i,j}$. Note that this graph is undirected since $s_{i,j} = s_{j,i}$; furthermore, if two instances i and j are unrelated, then they trivially satisfy $s_{i,j} = d$. Then, for any given problem instance j , we can optimize the subset of instances $\mathcal{Q}_j \subseteq [N]$ from which to transfer knowledge. For simplicity, we assume a strategy where we fix a threshold \tilde{s} , and keep all instances with sparse disparity at most \tilde{s} — i.e.,

$$\mathcal{Q}_j = \{i \in [N] \mid s_{i,j} \leq \tilde{s}\}.$$

We denote the effective number of instances by $\tilde{N} = |\mathcal{Q}_j|$. Under this assumption, there is a tradeoff between choosing smaller \tilde{s} , which yields smaller \tilde{N} (resulting in lower bias but larger variance), and larger \tilde{s} , which yields larger \tilde{N} (resulting in higher bias but smaller variance). The optimal choice of \tilde{s} (and correspondingly, \tilde{N}) depends on the relationship between \tilde{s} and \tilde{N} . Here we consider a natural power law scaling — i.e.,

$$\tilde{s} = \min(\tilde{N}^\alpha, d), \quad (22)$$

for some $\alpha \geq 0$. In other words, as we increase the number of neighbouring instances we include, our sparsity parameter increases by some power law \tilde{N}^α until it eventually hits the maximum possible value d . Our main result allows us to easily compute the optimal choice of \tilde{N} , resulting in regret bounds that scale with the network density α .

In this setting, we can find the optimal choice to be $\tilde{N} = d^{\frac{1}{\alpha+1}}$, resulting in the following upper bound on the cumulative regret at instance j ; note that it scales with the network density α .

COROLLARY 6. *Under the network structure given by (22) and when there are sufficient instances $N = \Omega(d^{\frac{1}{\alpha+1}})$, the optimal estimation error of $\hat{\beta}_{RM}^j$ for a single instance j using Algorithm 1 is*

$$\|\hat{\beta}_{RM}^j - \beta^j\|_1 = \tilde{O}\left(\frac{d^{\frac{2\alpha+1}{2(\alpha+1)}}}{\sqrt{n_j}}\right),$$

where we choose $\tilde{N} = \Theta(d^{\frac{1}{\alpha+1}})$.

Proof of Corollary 6 Our network structure is exogenous regarding the sparsity \tilde{s} ; thus, all previous analyses for Theorem 1 and Corollary 1 still go through for an arbitrary number of selected instances \tilde{N} . Plugging $\tilde{s} = \tilde{N}^\alpha$ in the error bound derived in Corollary 1 (note that s and N in Corollary 1 are now \tilde{s} and \tilde{N} in this setting) and optimizing \tilde{N} , we obtain the optimal value of $\tilde{N} = \Theta\left(d^{\frac{1}{\alpha+1}}\right)$. Finally, the constraint on δ becomes $\delta \geq \exp\left(-\frac{d^{\frac{1}{\alpha+1}}}{9}\left(\frac{C_0-2}{4C_0}\right)^2 + \log(6d)\right)$ given the above choice of \tilde{s} and \tilde{N} . \square

Again, we obtain an improvement in the context dimension d ; in particular, the error of **RMEstimator** scales in proportion to $d^{\frac{2\alpha+1}{2(\alpha+1)}}$, which is always smaller than the d -scaling of OLS where we do not learn from other instances. The extent of this estimation error scales with the network density α . When $\alpha \rightarrow 0$ (i.e., there are many instances with high similarity to the target instance), we eliminate a factor of \sqrt{d} , which can be substantial in high dimension and is aligned with Corollary 1; when $\alpha \rightarrow \infty$ (i.e., there are essentially no instances with high similarity to the target instance), our improvement disappears and the error converges to that of OLS.

Appendix B: Lower Bounds for Baselines

In this section, we provide detailed statements and proofs for the lower bounds discussed in §3.4 and Table 1. At a high level, our lower bounds follow by exhibiting a concrete instantiation of the parameters β^j and data (\mathbf{X}^j, Y^j) and establishing a lower bound on the error of the estimator for this instantiation. Recall that our error measure is defined in (8), i.e.,

$$\ell(\hat{\beta}^j, \beta^j) = \sup_{\mathcal{G}} \mathbb{E} \left[\|\hat{\beta}^j - \beta^j\|_1 \right],$$

where $\mathcal{G} = \{ \{\mathbf{X}^j\}_{j \in [N]}, \{\beta^j\}_{j \in [N]}, \{\mathcal{P}_\epsilon^j\}_{j \in [N]} \}$ satisfies Assumption 1, \mathcal{P}_ϵ^j is the distribution of the subgaussian noise ϵ^j , and the expectation is taken with respect to ϵ^j 's. Since this error measure takes a worst-case scenario over \mathcal{G} , it suffices to show the lower bound for a specific case where the assumptions hold.

For the remainder of this section, we assume $\epsilon^j \sim \mathcal{N}(\mathbf{0}, \sigma_j^2 \mathbf{I})$, and $\hat{\Sigma}^j = \mathbf{I}$ for $j \in [N]$. Our choices of errors ϵ^j are all gaussian, which ensures the parameter estimates are gaussian as well, thereby enabling us to obtain lower bounds by applying the following lemma:

LEMMA 5. *Consider a multivariate gaussian random variable $X \sim \mathcal{N}(\mu, \Sigma) \in \mathbb{R}^d$. We have*

$$\mathbb{E}[\|X\|_1] \geq \frac{1}{2} \|\mu\|_1 + \frac{1}{\sqrt{2\pi}} \text{tr}(\Sigma^{\frac{1}{2}}).$$

Proof of Lemma 5 Consider the i^{th} component of X , i.e., $X_{(i)}$. Let $\sigma_i^2 = \Sigma_{(i,i)}$. We have $X_{(i)} \sim \mathcal{N}(\mu_{(i)}, \sigma_i^2)$. Without loss of generality, assume $\mu_{(i)} \geq 0$; otherwise, we can consider $-X_{(i)}$ instead and its ℓ_1 norm stays the same. By our gaussian assumption, it holds that

$$\mathbb{E}[|X_{(i)}|] = \int_{-\infty}^{\infty} |x + \mu_{(i)}| \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{x^2}{2\sigma_i^2}} dx \geq \int_0^{\infty} (x + \mu_{(i)}) \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{x^2}{2\sigma_i^2}} dx = \frac{1}{2} \mu_{(i)} + \frac{1}{\sqrt{2\pi}} \sigma_i.$$

Then, we further have

$$\mathbb{E}[\|X\|_1] = \sum_{i \in [d]} \mathbb{E}[|X_{(i)}|] = \frac{1}{2} \|\mu\|_1 + \frac{1}{\sqrt{2\pi}} \sum_{i \in [d]} \sqrt{\Sigma_{(i,i)}} \geq \frac{1}{2} \|\mu\|_1 + \frac{1}{\sqrt{2\pi}} \text{tr}(\Sigma^{\frac{1}{2}}),$$

where the last step uses $\sqrt{\Sigma_{(i,i)}} = \|\Sigma_{(i,\cdot)}^{\frac{1}{2}}\|_2 \geq \Sigma_{(i,i)}^{\frac{1}{2}}$. \square

B.1. Independent Estimator

First, we consider the independent estimator $\hat{\beta}_{\text{ind}}^j$, which simply uses OLS independently on each instance (i.e., it does not perform any learning across instances). In particular, we have the following result:

PROPOSITION 7 (Full Statement of Proposition 1). *The estimation error of the independent estimator in both the standard and data-poor regimes satisfies*

$$\ell(\hat{\beta}_{\text{ind}}^j, \beta^j) = \Omega\left(\frac{d}{\sqrt{n_j}}\right).$$

Proof of Proposition 1 For our choice of \mathbf{X}^j and ϵ^j , the estimation error follows a gaussian distribution:

$$\hat{\beta}_{\text{ind}}^j - \beta^j \sim \mathcal{N}\left(\mathbf{0}, \frac{\sigma_j^2}{n_j} \mathbf{I}\right).$$

Therefore, using Lemma 5, we have

$$\mathbb{E}\left[\|\hat{\beta}_{\text{ind}}^j - \beta^j\|_1\right] \geq \frac{d\sigma_j}{\sqrt{2\pi n_j}}. \quad \square$$

B.2. Averaging/Pooling Estimator

Next, we consider the averaging estimator $\hat{\beta}_{\text{avg}}^j$ and the pooling estimator $\hat{\beta}_{\text{pool}}^j$. The averaging estimator simply takes the average of the independent parameter estimates across instances to reduce variance, while the pooling estimator pools all the data \mathbf{X}^j, Y^j across instances, and then uses OLS on this pooled dataset. Note that these two estimators do not vary across instances j 's. The averaging estimator is identical to Step 1 of the averaging multitask estimator described in §3.3. The pooling estimator, different from the averaging estimator, accounts for differences in the covariance matrices $\hat{\Sigma}^j = \mathbf{X}^{j\top} \mathbf{X}^j / n_j$ across instances.

PROPOSITION 8 (Full Statement of Proposition 2). *The estimation error of the averaging/pooling estimator in the standard regime satisfies*

$$\ell(\hat{\beta}_{\text{avg}}^j, \beta^j) = \Omega\left(\|\delta^j\|_1 + \frac{d}{\sqrt{N n_j}}\right),$$

and in the data-poor regime satisfies

$$\ell(\hat{\beta}_{\text{avg}}^j, \beta^j) = \Omega\left(\|\delta^j\|_1 + \frac{1}{\sqrt{N n_j}}\right).$$

Proof of Proposition 2 For our choice of \mathbf{X}^j 's and ϵ^j 's, the estimation error of the averaging estimator follows a gaussian distribution:

$$\hat{\beta}_{\text{avg}}^j - \beta^j = \frac{1}{N} \sum_{i \in [N]} (\hat{\beta}_{\text{ind}}^i - \beta^i) + \frac{1}{N} \sum_{i \in [N]} (\delta^i - \delta^j) \sim \mathcal{N}\left(\frac{1}{N} \sum_{i \in [N]} (\delta^i - \delta^j), \frac{1}{N^2} \sum_{i \in [N]} \frac{\sigma_i^2}{n_i} \mathbf{I}\right).$$

Therefore, by Lemma 5, we have

$$\mathbb{E}\left[\|\hat{\beta}_{\text{avg}}^j - \beta^j\|_1\right] \geq \frac{1}{2} \left\| \frac{1}{N} \sum_{i \in [N]} (\delta^i - \delta^j) \right\|_1 + \frac{1}{\sqrt{2\pi}} \sqrt{\frac{1}{N} \sum_{i \in [N]} \frac{\sigma_i^2 n_j}{n_i}} \frac{d}{\sqrt{N n_j}}.$$

Similarly, the estimation error of the pooling estimator also follows a gaussian distribution:

$$\begin{aligned} \hat{\beta}_{\text{pool}}^j - \beta^j &= \left(\sum_{i \in [N]} \mathbf{X}^{i\top} \mathbf{X}^i \right)^{-1} \left(\sum_{i \in [N]} \mathbf{X}^{i\top} \mathbf{X}^i (\delta^i - \delta^j) \right) + \left(\sum_{i \in [N]} \mathbf{X}^{i\top} \mathbf{X}^i \right)^{-1} \left(\sum_{i \in [N]} \mathbf{X}^{i\top} \epsilon^i \right) \\ &\sim \mathcal{N}\left(\frac{\sum_{i \in [N]} n_i (\delta^i - \delta^j)}{\sum_{i \in [N]} n_i}, \frac{\sum_{i \in [N]} \sigma_i^2 n_i}{(\sum_{i \in [N]} n_i)^2} \mathbf{I}\right). \end{aligned}$$

Therefore, Lemma 5 implies

$$\mathbb{E}\left[\|\hat{\beta}_{\text{pool}}^j - \beta^j\|_1\right] \geq \frac{1}{2} \left\| \frac{\sum_{i \in [N]} n_i (\delta^i - \delta^j)}{\sum_{i \in [N]} n_i} \right\|_1 + \frac{1}{\sqrt{2\pi}} \sqrt{\frac{(\sum_{i \in [N]} \sigma_i^2 n_i) N n_j}{(\sum_{i \in [N]} n_i)^2}} \frac{d}{\sqrt{N n_j}}.$$

For data-poor regime, we use all the instances except j to calculate $\hat{\beta}_{\text{avg}}^j$ and $\hat{\beta}_{\text{pool}}^j$. The proof is similar.

□

B.3. Averaging Multitask Estimator

Finally, we consider the averaging multitask estimator, which uses a traditional estimate of the mean instead of our robust estimate in Step 1. Our lower bound on the error of this estimator demonstrates the importance of robustness. Following the proof of the LASSO lower bound in Theorem 7.1 of Lounici et al. (2011), we assume that λ_j is chosen based on the corresponding upper bound analysis. Corresponding to Lemma 1, we take $\lambda_j = \sqrt{\frac{32\sigma_j^2}{n_j} \log(\frac{4d}{\delta})}$ and set $\delta = d^{-C}$ where $C \geq 3$ is a constant.

PROPOSITION 9 (Full Statement of Proposition 3). *Let the hyperparameter $\lambda_j = \sqrt{\frac{32(1+C)\sigma_j^2 \log(4d)}{n_j}}$, where $C \geq 3$ is a constant. Then, the estimation error of the averaging multitask estimator in the standard regime satisfies*

$$\ell(\hat{\beta}_{AM}^j, \beta^j) = \tilde{\Omega} \left(\frac{\min\{Ns, d\}}{\sqrt{n_j}} + \frac{d}{\sqrt{Nn_j}} \right),$$

and in the data-poor regime satisfies

$$\ell(\hat{\beta}_{AM}^j, \beta^j) = \tilde{\Omega} \left(\frac{\min\{Ns, d\}}{\sqrt{n_j}} \right).$$

Proof of Proposition 3 The proof strategy is adapted from that of Theorem 7.1 in Lounici et al. (2011). The first order condition of problem (4) is, for the i^{th} component,

$$\begin{cases} \frac{1}{n_j} \left(\mathbf{X}^{j\top} (Y^j - \mathbf{X}^j \hat{\beta}_{AM}^j) \right)_{(i)} = \lambda_j \text{sign}(\hat{\beta}_{AM,(i)}^j - \hat{\beta}_{AM,(i)}^\dagger) & \text{if } \hat{\beta}_{AM,(i)}^j \neq \hat{\beta}_{AM,(i)}^\dagger \\ \left| \frac{1}{n_j} \left(\mathbf{X}^{j\top} (Y^j - \mathbf{X}^j \hat{\beta}_{AM}^j) \right)_{(i)} \right| \leq \lambda_j & \text{if } \hat{\beta}_{AM,(i)}^j = \hat{\beta}_{AM,(i)}^\dagger. \end{cases} \quad (23)$$

Note that on the event \mathcal{H}^j in (15), it holds that $\frac{2}{n_j} |(\mathbf{X}^{j\top} \epsilon^j)_{(i)}| \leq \frac{\lambda_j}{2}$. Combining it with (23), we have

$$\frac{3\lambda_j}{4} \leq \left| \frac{1}{n_j} \left(\mathbf{X}^{j\top} (\mathbf{X}^j \beta^j - \mathbf{X}^j \hat{\beta}_{AM}^j) \right)_{(i)} \right| = |(\hat{\beta}_{AM}^j - \beta^j)_{(i)}|$$

for each i such that $\hat{\beta}_{AM,(i)}^j \neq \hat{\beta}_{AM,(i)}^\dagger$, where the last equality is from our assumption $\hat{\Sigma}^j = \mathbf{I}$. For the rest of the components, note that $\hat{\beta}_{AM,(i)}^j = \hat{\beta}_{AM,(i)}^\dagger$. Summing over all $i \in [d]$, we get

$$\|\hat{\beta}_{AM}^j - \beta^j\|_1 \geq \frac{3|\mathcal{V}|\lambda_j}{4} \quad (24)$$

with at least a probability of $\mathbb{P}[\mathcal{H}^j] \geq 1 - d^{-C}/2$ given our choice of λ_j , where $\mathcal{V} = \{i \in [d] \mid \hat{\beta}_{AM,(i)}^j \neq \hat{\beta}_{AM,(i)}^\dagger\}$. Define $\hat{\delta}_{AM}^j = \hat{\beta}_{AM}^j - \hat{\beta}_{AM}^\dagger$ and $\tilde{\delta}_{AM}^j = \beta^j - \tilde{\beta}_{AM}^\dagger$, where $\tilde{\beta}_{AM}^\dagger$ is defined in (5). Note that $|\mathcal{V}| = \|\hat{\delta}_{AM}^j\|_0$.

Now we want to construct a contradiction to show that $\hat{\delta}_{AM,(i)}^j = 0$ implies $\tilde{\delta}_{AM,(i)}^j = 0$ with high probability for any $i \in [d]$. If there exists $i \in [d]$ such that $\hat{\delta}_{AM,(i)}^j = 0$ but $\tilde{\delta}_{AM,(i)}^j \neq 0$. By the first order condition (23) and on the event \mathcal{H}^j , we have

$$|(\hat{\beta}_{AM}^\dagger - \beta^j)_{(i)}| = |(\hat{\beta}_{AM}^j - \beta^j)_{(i)}| \leq \frac{5\lambda_j}{4},$$

and hence

$$|(\beta^j - \tilde{\beta}_{AM}^\dagger)_{(i)}| \leq \frac{5\lambda_j}{4} + |(\tilde{\beta}_{AM}^\dagger - \hat{\beta}_{AM}^\dagger)_{(i)}|. \quad (25)$$

Note that

$$\hat{\beta}_{\text{AM}}^\dagger - \tilde{\beta}_{\text{AM}}^\dagger = \frac{1}{N} \sum_{i \in [N]} (\hat{\beta}_{\text{ind}}^i - \beta^i) \sim \mathcal{N} \left(\mathbf{0}, \frac{1}{N^2} \sum_{i \in [N]} \frac{\sigma_i^2}{n_i} \mathbf{I} \right).$$

Using a Chernoff bound, we have for any $t > 0$ and $i \in [d]$

$$\mathbb{P} \left[|(\hat{\beta}_{\text{AM}}^\dagger - \tilde{\beta}_{\text{AM}}^\dagger)_{(i)}| \geq t \right] \leq 2 \exp \left(- \frac{t^2}{\frac{2}{N^2} \sum_{m \in [N]} \frac{\sigma_m^2}{n_m}} \right).$$

Take $t = \sqrt{\frac{2C \log(4d)}{N^2} \sum_{m \in [N]} \frac{\sigma_m^2}{n_m}}$, where C is any constant such that $C \geq 3$. Then, combined with a union bound over \mathcal{H}^j , the true parameters $\{\beta^j\}_{j \in [N]}$ should satisfy

$$|(\beta^j - \tilde{\beta}_{\text{AM}}^\dagger)_{(i)}| \leq \sqrt{\frac{50(1+C)\sigma_j^2}{n_j} \log(4d)} + \sqrt{\frac{2C}{N^2} \sum_{m \in [N]} \frac{\sigma_m^2}{n_m} \log(4d)}$$

with at least a probability of $1 - d^{-C}$.

Since we consider a worst-case error over all possible $\{\beta^j\}_{j \in [N]}$ that satisfy our assumptions, we only need to focus on two special cases in the following two situations respectively: (i) $Ns \leq d$ and (ii) $Ns > d$. In particular,

- (i). when $Ns \leq d$, consider a case where (a) $|\delta_{(i)}^k| > \sqrt{\frac{50(1+C)\sigma_j^2 N^2}{n_j} \log(4d)} + \sqrt{2C \sum_{m \in [N]} \frac{\sigma_m^2}{n_m} \log(4d)}$ for any $i \in [d]$ and $k \in [N]$ when $\delta_{(i)}^k \neq 0$, and (b) $|\{k \in [N] \mid \delta_{(i)}^k \neq 0\}| \leq 1$ for any $i \in [d]$;
- (ii). when $Ns > d$, consider a case where (a) $\delta_{(i)}^j > \sqrt{\frac{50(1+C)\sigma_j^2 d^2}{s^2 n_j} \log(4d)} + \sqrt{\frac{2C d^2}{s^2 N^2} \sum_{m \in [N]} \frac{\sigma_m^2}{n_m} \log(4d)}$ and $\delta_{(i)}^k < - \left(\sqrt{\frac{50(1+C)\sigma_j^2 d^2}{s^2 n_j} \log(4d)} + \sqrt{\frac{2C d^2}{s^2 N^2} \sum_{m \in [N]} \frac{\sigma_m^2}{n_m} \log(4d)} \right)$ for any $k \neq j$ and for any $i \in [d]$ when $\delta_{(i)}^j, \delta_{(i)}^k \neq 0$, and (b) $|\{k \in [N] \mid \delta_{(i)}^k \neq 0\}| = Ns/d$ for any $i \in [d]$.

Both cases give rise to a contradiction. As a consequence, whenever $\hat{\delta}_{\text{AM},(i)}^j = 0$, it holds that $\tilde{\delta}_{\text{AM},(i)}^j = 0$ with probability at least $1 - d^{-C}$, and hence $\|\hat{\delta}_{\text{AM}}^j\|_0 \geq \|\tilde{\delta}_{\text{AM}}^j\|_0$. Correspondingly, note that $|\mathcal{V}| \geq \min\{Ns, d\}$ given $\|\tilde{\delta}_{\text{AM}}^j\|_0 = \min\{Ns, d\}$ in our design above.

Additionally, it always holds true that

$$\|\hat{\beta}_{\text{AM}}^j - \beta^j\|_1 \geq \|(\hat{\beta}_{\text{AM}}^\dagger - \beta^j)_{\mathcal{V}^c}\|_1.$$

By Lemma 5 and given the set \mathcal{V} , we have

$$\begin{aligned} \mathbb{E} \left[\|(\hat{\beta}_{\text{AM}}^\dagger - \beta^j)_{\mathcal{V}^c}\|_1 \right] &\geq \frac{1}{2} \left\| \frac{1}{N} \sum_{i \in [N]} (\delta^i - \delta^j)_{\mathcal{V}^c} \right\|_1 + \frac{1}{\sqrt{2\pi}} \sqrt{\frac{1}{N} \sum_{i \in [N]} \frac{\sigma_i^2 n_j}{n_i} \frac{|\mathcal{V}^c|}{\sqrt{N n_j}}} \\ &= \frac{1}{\sqrt{2\pi}} \sqrt{\frac{1}{N} \sum_{i \in [N]} \frac{\sigma_i^2 n_j}{n_i} \frac{|\mathcal{V}^c|}{\sqrt{N n_j}}}, \end{aligned} \quad (26)$$

where the last equality holds because the support of $\hat{\delta}_{\text{AM}}^j$ includes that of $\tilde{\delta}_{\text{AM}}^j$ as shown in the last paragraph. Given $|\mathcal{V}| \leq d$ and the fact that $|\mathcal{V}| \geq \min\{Ns, d\}$ holds with probability at least $1 - d^{-C}$, we derive from (26) that

$$\mathbb{E} \left[\|\hat{\beta}_{\text{AM}}^j - \beta^j\|_1 \right] \geq \mathbb{E} \left[\frac{1}{\sqrt{2\pi}} \sqrt{\frac{1}{N} \sum_{i \in [N]} \frac{\sigma_i^2 n_j}{n_i} \frac{d - |\mathcal{V}|}{\sqrt{N n_j}}} \middle| d \geq |\mathcal{V}| \geq \min\{Ns, d\} \right] (1 - d^{-C}). \quad (27)$$

Further, from (24), we also have

$$\mathbb{E} \left[\|\hat{\beta}_{\text{AM}}^j - \beta^j\|_1 \right] \geq \mathbb{E} \left[\frac{3|\mathcal{V}|\lambda_j}{4} \right] \mathbb{P} [\mathcal{H}^j] \geq \mathbb{E} \left[\frac{3(1-d^{-C}/2)|\mathcal{V}|\lambda_j}{4} \middle| d \geq |\mathcal{V}| \geq \min\{Ns, d\} \right] (1-d^{-C}). \quad (28)$$

Combining (27) and (28), we have

$$\mathbb{E} \left[\|\hat{\beta}_{\text{AM}}^j - \beta^j\|_1 \right] \geq \frac{1-d^{-C}}{2} \mathbb{E} \left[\frac{3(1-d^{-C}/2)|\mathcal{V}|\lambda_j}{4} + \frac{1}{\sqrt{2\pi}} \sqrt{\frac{1}{N} \sum_{i \in [N]} \frac{\sigma_i^2 n_j}{n_i} \frac{d-|\mathcal{V}|}{\sqrt{Nn_j}}} \middle| d \geq |\mathcal{V}| \geq \min\{Ns, d\} \right],$$

where we use $\max\{a, b\} \geq (a+b)/2$. As the above lower bound is linear in $|\mathcal{V}|$, its minimum value is taken at either end of the interval $[\min\{Ns, d\}, d]$. Therefore, we can derive that

$$\begin{aligned} \mathbb{E} \left[\|\hat{\beta}_{\text{AM}}^j - \beta^j\|_1 \right] &\geq (1-d^{-C}) \min \left\{ \frac{3(1-d^{-C}/2)d}{2} \sqrt{\frac{2(1+C)\sigma_j^2}{n_j} \log(4d)}, \right. \\ &\quad \left. \frac{3(1-d^{-C}/2) \min\{Ns, d\}}{2} \sqrt{\frac{2(1+C)\sigma_j^2}{n_j} \log(4d)} + \frac{1}{2\sqrt{2\pi}} \sqrt{\frac{1}{N} \sum_{i \in [N]} \frac{\sigma_i^2 n_j}{n_i} \frac{\max\{d-Ns, 0\}}{\sqrt{Nn_j}}} \right\}. \end{aligned}$$

Therefore, when $Ns = o(d)$, we can write

$$\mathbb{E} \left[\|\hat{\beta}_{\text{AM}}^j - \beta^j\|_1 \right] = \tilde{\Omega} \left(\frac{Ns}{\sqrt{n_j}} + \frac{d}{\sqrt{Nn_j}} \right);$$

when $Ns = \Omega(d)$, we get

$$\mathbb{E} \left[\|\hat{\beta}_{\text{AM}}^j - \beta^j\|_1 \right] = \tilde{\Omega} \left(\frac{d}{\sqrt{n_j}} \right).$$

The proof for the data-poor regime is similar, where in the first stage we use all the instances except j to calculate $\hat{\beta}_{\text{AM}}^{\dagger}$. \square

Appendix C: Multitask Bandits

In this section, we give proofs for Theorem 2, Corollary 3, Corollary 4 and Corollary 6 in §5. We list the hyperparameter choices for Theorem 2 as follows:

$$\begin{aligned} \omega_0 &= \zeta_0 + \eta_0, \quad \zeta_0 = \zeta_{1,0} = \frac{C_0 - 2}{4C_0} \sqrt{\frac{s}{d}}, \quad \eta_0 = \sqrt{\frac{27 \log(d) |\mathcal{B}_0|}{qN}}, \quad \eta_{1,0} = \sqrt{\frac{9}{\rho N}}, \\ \lambda_0 &= \max_{i \in [N]} \sqrt{\frac{96\sigma_i^2 x_{\max}^2 K \log(d) |\mathcal{B}_0|}{q}}, \quad \lambda_{1,0} = \max_{i \in [N]} \sqrt{\frac{384\sigma_i^2 x_{\max}^2}{p_*}}, \end{aligned}$$

and

$$q = \max \left\{ \frac{2 \cdot 384^3 C_0^2 x_{\max}^4 (\max_{i \in [N]} \sigma_i^2 / p_i) K s d \log(d)}{(C_0 - 2)^2 h^2 p_*^2 \psi^2}, \frac{576 C_0^2 x_{\max}^2 (\max_{i \in [N]} \sigma_i^2 / p_i) K s d \log(N)}{h^2 p_* \psi}, \right. \\ \left. \frac{6 \cdot 192^2 C_0^2 x_{\max}^2 (\max_{i \in [N]} \sigma_i^2 / p_i) K d^2 \log(d) \log(N)}{h^2 p_* \psi N}, \frac{96 x_{\max}^2 K d \log(dN)}{p_* \psi (\min_{i \in [N]} p_i)} \right\}.$$

C.1. Proof Strategy

First we sketch the proof of our main regret bound in Theorem 2. The proof builds on the regret analysis of LASSO Bandit (Bastani and Bayati 2020), but with the confidence intervals afforded by our robust multitask estimator provided in Theorem 1. As noted earlier, one key added challenge is the requirement that the OLS estimators $\{\hat{\beta}_{\text{ind}}^j\}_{j \in [N]}$ across different instances be independent in order to invoke our robust multitask estimator; RMBandit achieves this goal using a batching strategy, as highlighted in Lemma 6.

Forced-Sample Estimator. Our algorithm uses a separate forced-sample estimator, which we can guarantee is close to the true parameter with high probability.

PROPOSITION 10. When $N = \Omega(\log(d)\log(T))$, the forced-sample estimator $\hat{\beta}_{k,0}^j = \hat{\beta}_k^j(\mathcal{B}_0, \lambda_{0,j}, \omega_0)$ of instance j and arm k in Algorithm 2 satisfies

$$\mathbb{P}\left[\|\hat{\beta}_{k,0}^j - \beta_k^j\|_1 \geq \frac{h}{4x_{\max}}\right] \leq \frac{8}{T},$$

for the hyperparameter choices ζ_0 , η_0 , λ_0 , and q specified in Appendix C.

We give a proof in Appendix C.2. At a high level, this result follows directly from our tail inequality of a random design (i.e., Proposition 5), since the forced samples are i.i.d. random variables.

All-Sample Estimator. Next, we provide a tail inequality for our all-sample estimator for all arms that belong in \mathcal{K}_{opt}^j (it suffices to consider these arms since the forced-sample estimator is sufficiently accurate to exclude arms in \mathcal{K}_{sub}^j from consideration). In contrast to the forced-sample estimator, which is based on $\mathcal{O}(\log(T))$ samples, the all-sample estimator is based on $\mathcal{O}(T)$ samples (since we will show that all optimal arms receive a linear number of samples with high probability). Therefore, the all-sample estimator has smaller error than the forced-sample estimator (the tradeoff is that these samples are adaptively assigned to arms, so they may be collected from biased regions of the covariate space; thus, the i.i.d. samples generated when using the forced-sample estimator are needed to ensure that the all-sample estimator converges). In particular, define the following event, which says that all the forced-sample estimators have small error:

$$\mathcal{A} = \left\{ \|\hat{\beta}_k^j(\mathcal{B}_0, \lambda_{0,j}, \omega_0) - \beta_k^j\|_1 \leq \frac{h}{4x_{\max}}, \forall j \in [N], k \in [K] \right\}. \quad (29)$$

This event holds with high probability by Proposition 10. Our next result shows that our all-sample estimator satisfies the following tail inequality conditional on the event \mathcal{A} .

PROPOSITION 11. When \mathcal{A} holds and $N = \Omega(\log(d)\log(T))$, the all-sample estimator $\hat{\beta}_{k,\bar{m}}^j = \hat{\beta}_k^j(\mathcal{B}_{\bar{m}}, \lambda_{1,j,\bar{m}}, \omega_{1,m})$ of instance j and optimal arm $k \in \mathcal{K}_{opt}^j$ in Algorithm 2 satisfies

$$\|\hat{\beta}_{k,\bar{m}}^j - \beta_k^j\|_1 \leq C_1 \sqrt{\frac{sd \log(dp_j |\mathcal{B}_{\bar{m}}|)}{p_j |\mathcal{B}_{\bar{m}}|}} + C_2 \sqrt{\frac{sd \log(\rho N)}{p_j |\mathcal{B}_{\bar{m}}|}} + C_3 d \sqrt{\frac{\log(dp_j |\mathcal{B}_{\bar{m}}|) \log(\rho N)}{N p_j |\mathcal{B}_{\bar{m}}|}}$$

with probability at least $1 - \left(\frac{6}{\min_{i \in \mathcal{W}_k} p_i |\mathcal{B}_m|} + \frac{4}{p_j |\mathcal{B}_m|} + \sum_{i \in \mathcal{W}_k} 7d \exp\left(-\frac{p_i p_j \psi |\mathcal{B}_m|}{32d x_{\max}^2}\right) \right)$ for the hyperparameter choices $\zeta_{1,0}$, $\eta_{1,0}$, and $\lambda_{1,0}$ specified in Theorem 2, where the constants C_1 , C_2 and C_3 are provided in Appendix C.3.

We give a proof in Appendix C.3. As previously discussed, since our all-sample estimators are constructed using all available samples, they may not be independent across instances; however, the trimmed mean estimator in Step 1 of our RMEstimator requires that the OLS inputs are independent across instances. By using a batching strategy, we ensure that the samples from the same batch \mathcal{B}_m that we use to calculate the trimmed mean are independent. Note that in Step 2 we still use all the data of the target instance from $\mathcal{B}_{\bar{m}}$. In particular, we have the following lemma:

LEMMA 6. The samples assigned to arm k in batch \mathcal{B}_m (for any $m \geq 1$) are independent across bandit instances conditioned on $\mathcal{F}_{m-1} = \sigma(\{X_t, Z_t, Y_t\}_{t \in \mathcal{B}_{m-1}})$, the σ -algebra generated by the samples from \mathcal{B}_{m-1} .

We give a proof in Appendix C.3. Given this lemma, Proposition 11 follows instantly by applying Proposition 5.

Regret Bound. Finally, we describe how the above results enable us to prove Theorem 2. For this regret analysis, we group time steps $t \in [T]$ into three possible cases, and bound the regret across time steps in each case separately:

- (i). When $T \leq N$, or from the forced-sample batch ($t \in \mathcal{B}_0$) or the first batch using all-sample estimator ($t \in \mathcal{B}_1$);
- (ii). When $T \geq N$, from all the remaining batches ($t \in \mathcal{B}_m$ for $m > 1$) such that \mathcal{A} does not hold;
- (iii). When $T \geq N$, from all the remaining batches ($t \in \mathcal{B}_m$ for $m > 1$) such that \mathcal{A} holds.

For Case (i), note that the sizes of the first two batches \mathcal{B}_0 and \mathcal{B}_1 are both $q \log(T)$ and q scales as $\tilde{\mathcal{O}}(Kd(sN + d))$. In the worst case, the regret for one time step is at most $2bx_{\max}$, so the regret in this case is bounded. For Case (ii), we have shown that the event \mathcal{A} holds with high probability. Similar to before, in the worst case, the regret for one time step is at most $2bx_{\max}$, so the regret in this case is bounded with high probability. Finally, for Case (iii) when \mathcal{A} holds, Proposition 11 guarantees that the all-sample estimator has small error with high probability, again ensuring that the regret is bounded with high probability. Details are provided in Appendix C.4.

C.2. Forced-Sample Estimator

Now we prove Proposition 10 that our forced sample estimators have small estimation errors with high probability. Our hyperparameter choices for forced-sample estimators include ζ_0 and η_0 listed at the beginning of §C, plus $\lambda_{0,j} = \lambda_0 / \sqrt{|\mathcal{B}_0^j|}$, as in Algorithm 2. For simplicity, we use ζ , η and λ_j to represent ζ_0 , η_0 and $\lambda_{0,j}$ respectively in the following.

Additional Notation . We start by defining some additional notation. Let \mathcal{B}_0^j be the index set of those observed at instance j , $\mathcal{B}_{0,k}^j$ be the subset forced sampled at arm k and $\bar{\mathcal{B}}_{0,k}^j$ be the subset of all $t \in \mathcal{B}_{0,k}^j$ such that $X_t \in U_k^j$; in particular,

$$\begin{aligned} \mathcal{B}_0^j &= \{t \in \mathcal{B}_0 \mid Z_t = j\}, \quad \mathcal{B}_{0,k}^j = \left\{ t \in \mathcal{B}_0 \mid Z_t = j, (k-1) \equiv \left(\sum_{r \in [t]} \mathbb{1}(Z_r = j) - 1 \right) \bmod K \right\}, \\ \bar{\mathcal{B}}_{0,k}^j &= \left\{ t \in \mathcal{B}_0 \mid Z_t = j, X_t \in U_k^{Z_t}, (k-1) \equiv \left(\sum_{r \in [t]} \mathbb{1}(Z_r = j) - 1 \right) \bmod K \right\}. \end{aligned}$$

We use $\widehat{\Sigma}(\mathcal{B})$ to represent the sample covariance matrix calculated using the samples $\{X_t\}_{t \in \mathcal{B}}$, i.e., $\widehat{\Sigma}(\mathcal{B}) = \sum_{t \in \mathcal{B}} X_t X_t^\top / |\mathcal{B}|$.

LEMMA 7. *The forced samples at arm k are independent across bandit instances.*

Proof of Lemma 7 The forced samples of arm k at instance j are $\{(X_t, Y_t)\}_{t \in \mathcal{B}_{0,k}^j}$, where the set of covariates is

$$\left\{ X_t \mid t \in \mathcal{B}_{0,k}^j, Z_t = j, (k-1) \equiv \left(\sum_{r \in [t]} \mathbb{1}(Z_r = j) - 1 \right) \bmod K \right\}.$$

Since $\{(X_t, Z_t)\}_{t \in \mathcal{B}_0}$ are independent, X_t is independent of $Z_{t'}$ and $X_{t'}$ conditional on Z_t for any $t' \neq t$ and $t' \in \mathcal{B}_0$. Therefore, given $Z_t = j$, we derive that $\sum_{r \in [t]} \mathbb{1}(Z_r = j) - 1 = \sum_{r \in [t-1]} \mathbb{1}(Z_r = j)$ is also independent of X_t . This implies X_t 's observed at arm k are independent across bandit instances. Similarly, Y_t 's are also independent across different instances noting that the noises ϵ_t only depends on Z_t by design. The result then follows. \square

LEMMA 8. *The samples $\{X_t\}_{t \in \mathcal{B}_{0,k}^j}$ are i.i.d. with distribution \mathcal{P}_X^j , and its subset $\{X_t\}_{t \in \bar{\mathcal{B}}_{0,k}^j}$ are i.i.d. with distribution $\mathcal{P}_{X|X \in U_k^j}^j$.*

Proof of Lemma 8 Similar to the proof of Lemma 7, we can show that $\{X_t\}_{t \in \mathcal{B}_{0,k}^j}$ are independent. As $\sum_{r \in [t]} \mathbb{1}(Z_r = j)$ is independent of X_t given $Z_t = j$, X_t follows the distribution \mathcal{P}_X^j . On the other hand, the subsamples $\{X_t\}_{t \in \bar{\mathcal{B}}_{0,k}^j}$ form the set

$$\left\{ X_t \mid t \in \bar{\mathcal{B}}_{0,k}^j, X_t \in U_k^{Z_t}, Z_t = j, (k-1) \equiv \left(\sum_{r \in [t]} \mathbb{1}(Z_r = j) - 1 \right) \bmod K \right\}.$$

Since $\{(X_t, Z_t)\}_{t \in \mathcal{B}_0}$ are independent, X_t is independent of $Z_{t'}$ and $X_{t'}$ conditional on $\{X_t \in U_k^{Z_t}, Z_t = j\}$ for any $t' \neq t$ and $t' \in \mathcal{B}_0$. Correspondingly, we can conclude that $\{X_t\}_{t \in \bar{\mathcal{B}}_{0,k}^j}$ are i.i.d. drawn from $\mathcal{P}_{X|X \in U_k^j}^j$. \square

LEMMA 9. *Define the event*

$$\bar{\mathcal{E}}_{0,k}^j = \left\{ \lambda_{\min}(\hat{\Sigma}(\bar{\mathcal{B}}_{0,k}^j)) \geq \frac{\psi}{2} \right\}. \quad (30)$$

Then, given $|\bar{\mathcal{B}}_{0,k}^j|$, we have

$$\mathbb{P}[\bar{\mathcal{E}}_{0,k}^j] \geq 1 - d \exp\left(-\frac{\psi |\bar{\mathcal{B}}_{0,k}^j|}{8dx_{\max}^2}\right).$$

Proof of Lemma 9 Note that $\{X_t X_t^\top\}_{t \in \bar{\mathcal{B}}_{0,k}^j}$ are i.i.d. according to Lemma 8. The rest of the proof follows that of Lemma 2. \square

LEMMA 10. *For any sets $\mathcal{B}, \bar{\mathcal{B}}$ with $\bar{\mathcal{B}} \subseteq \mathcal{B}$, it holds that $\lambda_{\min}(\hat{\Sigma}(\mathcal{B})) \geq \lambda_{\min}(\hat{\Sigma}(\bar{\mathcal{B}}))|\bar{\mathcal{B}}|/|\mathcal{B}|$.*

Proof of Lemma 10 See Lemma EC.23 in Bastani and Bayati (2020). \square

LEMMA 11. *Define the event*

$$\bar{\mathcal{D}}_{0,k}^j = \left\{ |\bar{\mathcal{B}}_{0,k}^j| \geq \frac{p_*}{2} |\mathcal{B}_{0,k}^j| \right\}. \quad (31)$$

Then, given $|\mathcal{B}_{0,k}^j|$, it holds that

$$\mathbb{P}[\bar{\mathcal{D}}_{0,k}^j] \geq 1 - 2 \exp\left(-\frac{p_* |\mathcal{B}_{0,k}^j|}{10}\right).$$

Proof of Lemma 11 Applying Lemma 35 to the indicator random variables $\mathbb{1}(t \in \bar{\mathcal{B}}_{0,k}^j)$ for all $t \in \mathcal{B}_{0,k}^j$ with $\mu = \mathbb{E}\left[\sum_{t \in \mathcal{B}_{0,k}^j} \mathbb{1}(t \in \bar{\mathcal{B}}_{0,k}^j)\right] = \sum_{t \in \mathcal{B}_{0,k}^j} \mathbb{P}[X_t \in U_k^j \mid Z_t = j]$, we have

$$\mathbb{P}\left[|\bar{\mathcal{B}}_{0,k}^j| - \mu \geq \frac{\mu}{2}\right] \leq 2 \exp\left(-\frac{\mu}{10}\right).$$

Noting $\mu \geq p_* |\mathcal{B}_{0,k}^j|$ by Assumption 6, the result then follows. \square

LEMMA 12. *Define the event*

$$\mathcal{M}_0^j = \left\{ |\mathcal{B}_0^j| \geq \frac{p_j}{2} |\mathcal{B}_0| \right\}. \quad (32)$$

Then, it holds that

$$\mathbb{P}[\mathcal{M}_0^j] \geq 1 - 2 \exp\left(-\frac{p_j |\mathcal{B}_0|}{10}\right).$$

Proof of Lemma 12 Applying Lemma 35 to the indicator random variables $\mathbb{1}(Z_t = j)$ for all $t \in \mathcal{B}_0$ with $\mu = \mathbb{E}[\sum_{t \in \mathcal{B}_0} \mathbb{1}(Z_t = j)] = \sum_{t \in \mathcal{B}_0} \mathbb{P}[Z_t = j] = p_j |\mathcal{B}_0|$, we have

$$\mathbb{P}\left[\left||\mathcal{B}_0^j| - p_j |\mathcal{B}_0|\right| \geq \frac{p_j |\mathcal{B}_0|}{2}\right] \leq 2 \exp\left(-\frac{p_j |\mathcal{B}_0|}{10}\right),$$

which implies our result. \square

LEMMA 13. *Given $|\mathcal{B}_{0,k}^j|$, we have*

$$\mathbb{P}\left[\lambda_{\min}(\widehat{\Sigma}(\mathcal{B}_{0,k}^j)) \geq \frac{p_* \psi}{4}\right] \geq 1 - \left(d \exp\left(-\frac{p_* \psi |\mathcal{B}_{0,k}^j|}{16 d x_{\max}^2}\right) + 2 \exp\left(-\frac{p_* |\mathcal{B}_{0,k}^j|}{10}\right)\right).$$

Proof of Lemma 13 Conditioned on $\bar{\mathcal{D}}_{0,k}^j$ in (31) and $\bar{\mathcal{E}}_{0,k}^j$ in (30), Lemma 10 implies $\lambda_{\min}(\widehat{\Sigma}(\mathcal{B}_{0,k}^j)) \geq p_* \psi / 4$. Therefore, we have

$$\mathbb{P}\left[\lambda_{\min}(\widehat{\Sigma}(\mathcal{B}_{0,k}^j)) \leq \frac{p_* \psi}{4}\right] \leq \mathbb{P}[(\bar{\mathcal{D}}_{0,k}^j)^c \cup (\bar{\mathcal{E}}_{0,k}^j)^c] \leq \mathbb{P}[(\bar{\mathcal{E}}_{0,k}^j)^c \mid \bar{\mathcal{D}}_{0,k}^j] + \mathbb{P}[(\bar{\mathcal{D}}_{0,k}^j)^c].$$

Using Lemma 11, the second term above is upper bounded by

$$\mathbb{P}[(\bar{\mathcal{D}}_{0,k}^j)^c] \leq 2 \exp\left(-\frac{p_* |\mathcal{B}_{0,k}^j|}{10}\right),$$

while Lemma 9 upper bounds the first term by

$$\mathbb{P}[(\bar{\mathcal{E}}_{0,k}^j)^c \mid \bar{\mathcal{D}}_{0,k}^j] \leq \mathbb{E}\left[d \exp\left(-\frac{\psi |\bar{\mathcal{B}}_{0,k}^j|}{8 d x_{\max}^2}\right) \mid \bar{\mathcal{D}}_{0,k}^j\right] \leq d \exp\left(-\frac{p_* \psi |\mathcal{B}_{0,k}^j|}{16 d x_{\max}^2}\right). \quad \square$$

Now we are ready to prove Proposition 10 using Proposition 5.

Proof of Proposition 10 Based on Lemma 7 and 8, the forced-sample OLS estimators are subgaussian and independent across instances so the conditions of Proposition 5 are satisfied. Applying Proposition 5 by setting $\phi = p_* \psi / 4$, we have

$$\begin{aligned} \mathbb{P}\left[\|\widehat{\beta}_{k,0}^j - \beta_k^j\|_1 \geq \frac{24 \lambda_j s}{p_* \zeta \psi} + 2 C_0 d (3\zeta + 4\eta) \max_{i \in [N]} \sqrt{\frac{\sigma_i^2}{p_* \psi |\mathcal{B}_{0,k}^i|}} \log\left(\frac{3}{\eta}\right)\right] \\ \leq 3d \exp\left(-\frac{N\eta^2}{9}\right) + 2d \exp\left(-\frac{\lambda_j^2 |\mathcal{B}_{0,k}^j|}{32 \sigma_j^2 x_{\max}^2}\right) + \sum_{i \in [N]} \mathbb{P}\left[\lambda_{\min}(\widehat{\Sigma}(\mathcal{B}_{0,k}^i)) \leq \frac{p_* \psi}{4}\right] \\ \leq 3d \exp\left(-\frac{N\eta^2}{9}\right) + 2d \exp\left(-\frac{\lambda_j^2 |\mathcal{B}_{0,k}^j|}{32 \sigma_j^2 x_{\max}^2}\right) \\ + \sum_{i \in [N]} d \exp\left(-\frac{p_* \psi |\mathcal{B}_{0,k}^i|}{16 d x_{\max}^2}\right) + \sum_{i \in [N]} 2 \exp\left(-\frac{p_* |\mathcal{B}_{0,k}^i|}{10}\right), \end{aligned}$$

where we use Lemma 13. By our forced sampling design, we have $|\mathcal{B}_{0,k}^j| = |\mathcal{B}_0^j|/K$. Plugging in, we have

$$\begin{aligned} \mathbb{P} \left[\|\widehat{\beta}_{k,0}^j - \beta_k^j\|_1 \geq \frac{24\lambda_j s}{p_* \zeta \psi} + 2C_0 d(3\zeta + 4\eta) \max_{i \in [N]} \sqrt{\frac{K\sigma_i^2}{p_* \psi |\mathcal{B}_0^i|}} \log\left(\frac{3}{\eta}\right) \right] \\ \leq 3d \exp\left(-\frac{N\eta^2}{9}\right) + 2d \exp\left(-\frac{\lambda_j^2 |\mathcal{B}_0^j|}{32K\sigma_j^2 x_{\max}^2}\right) \\ + \sum_{i \in [N]} d \exp\left(-\frac{p_* \psi |\mathcal{B}_0^i|}{16Kdx_{\max}^2}\right) + \sum_{i \in [N]} 2 \exp\left(-\frac{p_* |\mathcal{B}_0^i|}{10K}\right). \quad (33) \end{aligned}$$

Now we configure the hyperparameters ζ , λ_j , η and q (recall that $|\mathcal{B}_0| = q \log(T)$) such that our forced-sample estimator of arm k and instance j has estimation error smaller than $h/(4x_{\max})$ with high probability. Similar to the proof of Corollary 1, we take $\zeta = \frac{C_0-2}{4C_0} \sqrt{\frac{s}{d}}$. We further set

$$\eta = \sqrt{\frac{27 \log(d) |\mathcal{B}_0|}{qN}}, \quad \lambda_j = \sqrt{\frac{96 \sigma_j^2 x_{\max}^2 K \log(d) |\mathcal{B}_0|}{q |\mathcal{B}_0^j|}}.$$

In the following, we will frequently use the inequality

$$3 \log(T) \log(x) \geq \log(Tx) \quad (34)$$

for $T > 1$ and any $x > 1$. Given our choices of η and λ_j and inequality (34), the sum of the first two probabilities on the RHS of (33) is upper bounded by $5/T$. Further, using a union bound over $\bigcap_{i \in [N]} \mathcal{M}_0^i$ in (32) by Lemma 12, we have

$$\begin{aligned} \mathbb{P} \left[\|\widehat{\beta}_{k,0}^j - \beta_k^j\|_1 \geq \frac{768C_0\sigma_j x_{\max}}{(C_0-2)p_* \psi} \sqrt{\frac{3Ksd \log(d)}{qp_j}} + C_0 \left(\frac{3\sqrt{sd}}{2} + 24d \sqrt{\frac{3 \log(d) |\mathcal{B}_0|}{qN}} \right) \max_{i \in [N]} \sqrt{\frac{K\sigma_i^2 \log(N)}{2p_* \psi p_i |\mathcal{B}_0|}} \right] \\ \leq \frac{5}{T} + \sum_{i \in [N]} d \exp\left(-\frac{p_* \psi p_i |\mathcal{B}_0|}{32Kdx_{\max}^2}\right) + \sum_{i \in [N]} 4 \exp\left(-\frac{p_* p_i |\mathcal{B}_0|}{20K}\right) \\ \leq \frac{5}{T} + \sum_{i \in [N]} 3d \exp\left(-\frac{p_* \psi p_i |\mathcal{B}_0|}{32Kdx_{\max}^2}\right), \quad (35) \end{aligned}$$

where we use $\log(\frac{3}{\eta}) \leq \frac{\log(N)}{2}$, the first inequality uses $\mathbb{P}[(\mathcal{M}_0^i)^c] \leq 2 \exp(-\frac{p_* p_i |\mathcal{B}_0|}{20K})$, and the last inequality uses $\psi \leq \lambda_{\min}(\Sigma_k^j) \leq \lambda_{\max}(\Sigma_k^j) \leq \|X_t\|_2^2 \leq dx_{\max}^2$. Next, we choose a sufficiently large q such that

$$\begin{aligned} \frac{h}{8x_{\max}} &\geq \frac{768C_0\sigma_j x_{\max}}{(C_0-2)p_* \psi} \sqrt{\frac{3Ksd \log(d)}{qp_j}}, \quad \frac{h}{16x_{\max}} \geq C_0 \frac{3\sqrt{sd}}{2} \max_{i \in [N]} \sqrt{\frac{K\sigma_i^2 \log(N)}{2p_* \psi p_i |\mathcal{B}_0|}}, \\ \frac{h}{16x_{\max}} &\geq 24C_0d \sqrt{\frac{3 \log(d) |\mathcal{B}_0|}{qN}} \max_{i \in [N]} \sqrt{\frac{K\sigma_i^2 \log(N)}{2p_* \psi p_i |\mathcal{B}_0|}}, \quad \frac{3}{T} \geq \sum_{i \in [N]} 3d \exp\left(-\frac{p_* \psi p_i |\mathcal{B}_0|}{32Kdx_{\max}^2}\right). \end{aligned}$$

With a choice of q satisfying all the above constraints, the result then follows (35). To meet all the above conditions for all $j \in [N]$, it suffices to set

$$q = \max \left\{ \frac{2 \cdot 384^3 C_0^2 x_{\max}^4 (\max_{i \in [N]} \sigma_i^2 / p_i) K s d \log(d)}{(C_0-2)^2 h^2 p_*^2 \psi^2}, \frac{576 C_0^2 x_{\max}^2 (\max_{i \in [N]} \sigma_i^2 / p_i) K s d \log(N)}{h^2 p_* \psi}, \right. \\ \left. \frac{6 \cdot 192^2 C_0^2 x_{\max}^2 (\max_{i \in [N]} \sigma_i^2 / p_i) K d^2 \log(d) \log(N)}{h^2 p_* \psi N}, \frac{96 x_{\max}^2 K d \log(dN)}{p_* \psi (\min_{i \in [N]} p_i)} \right\}.$$

Remember we also require $\eta \leq 1/2 - 1/C_0 - \zeta$ according to Proposition 5, which is satisfied as long as

$$\log(T) \leq \left(\frac{C_0-2}{2C_0} \left(1 - \frac{1}{2} \sqrt{\frac{s}{d}} \right) \right)^2 \frac{N}{27 \log(d)}. \quad \square \quad (36)$$

C.3. All-Sample Estimator

Next, we prove Proposition 11, which shows our all-sample estimators have small errors with high probability.

The constants in the statement of Proposition 11 are

$$C_1 = \frac{96^2 C_0 \sigma_j x_{\max}}{(C_0 - 2)p_*^{3/2}\psi}, \quad C_2 = \frac{3C_0}{2p_*} \left(\max_{i \in [N]} \sqrt{\frac{2\sigma_i^2 p_j}{\psi p_i}} \right), \quad C_3 = \frac{24C_0}{p_*} \left(\max_{i \in [N]} \sqrt{\frac{2\sigma_i^2 p_j}{\psi p_i}} \right),$$

and our hyperparameter choices for all-sample estimators are

$$\zeta_{1,m} = \zeta_{1,0}, \quad \eta_{1,m} = \eta_{1,0} \sqrt{\log(d \min_{i \in [N], |\mathcal{B}_m^i| > 0} |\mathcal{B}_m^i|)}, \quad \lambda_{1,j,\bar{m}} = \lambda_{1,0} \sqrt{\frac{\log(d|\mathcal{B}_m^j|)}{|\mathcal{B}_m^j|}}$$

as in Algorithm 2. For simplicity, we use ζ , η and λ_j to represent $\zeta_{1,m}$, $\eta_{1,m}$ and $\lambda_{1,j,\bar{m}}$ respectively in the following.

Additional Notation. We begin with some additional notation. Let \mathcal{B}_m^j be the index set of those observed at instance j in batch m , $\mathcal{B}_{m,k}^j$ be the subset batch sampled at arm k , and $\bar{\mathcal{B}}_{m,k}^j$ be the subset of all $t \in \mathcal{B}_m^j$ such that $X_t \in U_k^j$ when \mathcal{A} holds; in particular, for any $m \geq 1$,

$$\mathcal{B}_m^j = \{t \in \mathcal{B}_m \mid Z_t = j\}, \quad \mathcal{B}_{m,k}^j = \{t \in \mathcal{B}_m \mid Z_t = j, \pi_{m-1}^{Z_t}(X_t) = k\}, \quad \bar{\mathcal{B}}_{m,k}^j = \{t \in \mathcal{B}_m \mid Z_t = j, X_t \in U_k^{Z_t}, \mathcal{A}\},$$

where, with a slight abuse of notation, we use $\pi_{m-1}^{Z_t}$ to represent the sub-policy of instance j estimated using the data from \mathcal{B}_{m-1} . More specifically, we use the notation \bar{m} to represent the union $\bigcup_{l=0}^m$ and \tilde{m} to represent $\bigcup_{l=1}^m$, e.g., $\mathcal{B}_{\bar{m}} = \bigcup_{l=1}^m \mathcal{B}_l$, $\mathcal{B}_{\tilde{m},k}^j = \bigcup_{l=1}^m \mathcal{B}_{l,k}^j$.

Then, we provide a bound on the probability of event \mathcal{A} in (29), which says that all the forced sample estimators have small errors based on our design.

LEMMA 14. *The event \mathcal{A} holds with at least a probability of $1 - 8KN/T$.*

Proof of Lemma 14 The result follows by applying a union bound over all arms and bandit instances using Proposition 10. \square

Next, we prove Lemma 6, which says that the samples collected at arm k in batch \mathcal{B}_m (for any $m \geq 1$) are conditionally independent across bandit instances.

Proof of Lemma 6 The collected samples of arm k at instance j in the batch \mathcal{B}_m are $\{(X_t, Y_t)\}_{t \in \mathcal{B}_{m,k}^j}$, where the set of covariates is

$$\{X_t \mid t \in \mathcal{B}_{m,k}^j, Z_t = j, \pi_{m-1}^{Z_t}(X_t) = k\}.$$

Note that our estimated policy $\pi_{m-1}^{Z_t}$ depends on Z_t and is constructed using samples from \mathcal{B}_{m-1} . Since $\{(X_t, Z_t, Y_t)\}_{t \in \mathcal{B}_m}$ are independent, $\{(X_t, Z_t, \pi_{m-1}^{Z_t}(X_t))\}_{t \in \mathcal{B}_m}$ are independent conditional on \mathcal{F}_{m-1} . Thus, for any $t' \neq t$ and $t' \in \mathcal{B}_m$, X_t is independent of $Z_{t'}$, $X_{t'}$ and $\pi_{m-1}^{Z_{t'}}(X_{t'})$ conditional on $\{Z_t = j, \pi_{m-1}^{Z_t}(X_t) = k, \mathcal{F}_{m-1}\}$. This implies X_t 's of arm k in batch m are conditionally independent across instances. Moreover, since the noises ϵ_t 's are independent of X_t 's and only depends on Z_t 's by design, the result then follows. \square

REMARK 7. Note that the samples across instances are also independent given $\{\mathcal{A}, \mathcal{F}_{m-1}\}$ since $\mathcal{A} \in \mathcal{F}_{m-1}$.

LEMMA 15. *For any $j \in [N]$, $k \in [K]$ and $t \notin \mathcal{B}_0$, if $Z_t = j$, $X_t \in U_k^j$ and the event \mathcal{A} holds, then Algorithm 2 plays the optimal arm k of instance j at time t based on the forced-sample estimator $\hat{\beta}_{k,0}^j$.*

Proof of Lemma 15 The proof is similar to Lemma EC.11 in Bastani and Bayati (2020). \square

LEMMA 16. (i) $\bar{\mathcal{B}}_{m,k}^j \subseteq \mathcal{B}_{m,k}^j$, (ii) $\{X_t\}_{t \in \mathcal{B}_{m,k}^j}$ are i.i.d. from $\mathcal{P}_{X|\pi_{m-1}^j(X)=k}^j$ conditioned on \mathcal{F}_{m-1} , and its subset $\{X_t\}_{t \in \bar{\mathcal{B}}_{m,k}^j}$ are i.i.d. from $\mathcal{P}_{X|X \in U_k^j}^j$, and (iii) $\{X_t\}_{t \in \bar{\mathcal{B}}_{m,k}^j}$ are i.i.d. from $\mathcal{P}_{X|X \in U_k^j}^j$.

Proof of Lemma 16 The first claim follows the result of Lemma 15. If $Z_t = j$, $X_t \in U_k^j$ and the event \mathcal{A} holds, then $\pi_{m-1}^{Z_t}(X_t) = k$ and hence $t \in \mathcal{B}_{m,k}^j$.

Similar to the proof of Lemma 6, we can show that $\{X_t\}_{t \in \mathcal{B}_{m,k}^j}$ are i.i.d. from distribution $\mathcal{P}_{X|\pi_{m-1}^j(X)=k}^j$ conditioned on \mathcal{F}_{m-1} . Additionally, note that the event \mathcal{A} only depends on forced samples from \mathcal{B}_0 and is hence independent of $\{(X_t, Z_t)\}_{t \in \mathcal{B}_m}$ for any $m \geq 1$. Therefore, X_t for any $t \in \bar{\mathcal{B}}_{m,k}^j$ follows distribution $\mathcal{P}_{X|X \in U_k^j}^j$. Since $\{(X_t, Z_t)\}_{t \in \mathcal{B}_m}$ are independent, X_t is independent of $Z_{t'}$ and $X_{t'}$ given $\{Z_t = j, X_t \in U_k^j, \mathcal{A}\}$ for any $t' \neq t$ and $t' \in \mathcal{B}_m$. Thus, $\{X_t\}_{t \in \bar{\mathcal{B}}_{m,k}^j}$ are also independent.

Correspondingly, we can show $\{X_t\}_{t \in \bar{\mathcal{B}}_{m,k}^j}$ are also i.i.d. from $\mathcal{P}_{X|X \in U_k^j}^j$, where $\bar{\mathcal{B}}_{m,k}^j = \bigcup_{l=1}^m \bar{\mathcal{B}}_{l,k}^j = \{t \in \bigcup_{l=1}^m \mathcal{B}_l \mid Z_t = j, X_t \in U_k^j, \mathcal{A}\}$. \square

REMARK 8. Note that (ii) and (iii) both hold further conditioned on \mathcal{A} . The first statement in (ii) still holds as $\mathcal{A} \in \mathcal{F}_{m-1}$ while the second statement in (ii) and the statement in (iii) hold as $\{(X_t, Z_t, Y_t)\}_{t \in \mathcal{B}_0}$ are independent of $\{(X_t, Z_t)\}_{t \in \mathcal{B}_m}$.

LEMMA 17. Define the event

$$\bar{\mathcal{D}}_{m,k}^j = \left\{ |\bar{\mathcal{B}}_{m,k}^j| \geq \frac{p_*}{2} |\mathcal{B}_m^j| \right\}. \quad (37)$$

Then, given $|\mathcal{B}_m^j|$, it holds that

$$\mathbb{P}[\bar{\mathcal{D}}_{m,k}^j \mid \mathcal{A}] \geq 1 - 2 \exp\left(-\frac{p_* |\mathcal{B}_m^j|}{10}\right).$$

Proof of Lemma 17 By definition of $\bar{\mathcal{B}}_{m,k}^j$, we have

$$|\bar{\mathcal{B}}_{m,k}^j| = \sum_{t \in \mathcal{B}_m^j} \mathbb{1}(t \in \bar{\mathcal{B}}_{m,k}^j) = \sum_{t \in \mathcal{B}_m^j} \mathbb{1}(Z_t = j, X_t \in U_k^j) \mathbb{1}(\mathcal{A}).$$

Then, take

$$\mu = \mathbb{E}[|\bar{\mathcal{B}}_{m,k}^j| \mid \mathcal{A}] = \mathbb{E}\left[\sum_{t \in \mathcal{B}_m^j} \mathbb{1}(Z_t = j, X_t \in U_k^j)\right] = \sum_{t \in \mathcal{B}_m^j} \mathbb{P}[X_t \in U_k^j \mid Z_t = j],$$

where the second equality is from the fact that $\{(X_t, Z_t)\}_{t \in \mathcal{B}_m}$ are independent of $\{(X_t, Z_t, Y_t)\}_{t \in \mathcal{B}_0}$. Using Lemma 35 and $\mu \geq p_* |\mathcal{B}_m^j|$ given Assumption 6, we can then get our result. \square

LEMMA 18. Define the events

$$\mathcal{M}_m^j = \left\{ |\mathcal{B}_m^j| \geq \frac{p_j}{2} |\mathcal{B}_m| \right\}, \quad \mathcal{M}_{\bar{m}}^j = \left\{ \frac{3p_j}{2} |\mathcal{B}_{\bar{m}}| \geq |\mathcal{B}_m^j| \geq \frac{p_j}{2} |\mathcal{B}_{\bar{m}}| \right\}. \quad (38)$$

Then, we have

$$\mathbb{P}[\mathcal{M}_m^j] \geq 1 - 2 \exp\left(-\frac{p_j |\mathcal{B}_m|}{10}\right), \quad \mathbb{P}[\mathcal{M}_{\bar{m}}^j] \geq 1 - 2 \exp\left(-\frac{p_j |\mathcal{B}_{\bar{m}}|}{10}\right).$$

Proof of Lemma 18 The result follows a similarly proof strategy of Lemma 12. \square

REMARK 9. Note that the high probability bound of $|\mathcal{B}_m^j|$ also holds conditioned on \mathcal{A} as $\{Z_t\}_{t \in \mathcal{B}_m}$ are independent of $\{(X_t, Z_t, Y_t)\}_{t \in \mathcal{B}_0}$.

LEMMA 19. *Given $|\mathcal{B}_m^j|$, we have*

$$\mathbb{P} \left[\lambda_{\min}(\widehat{\Sigma}(\mathcal{B}_{m,k}^j)) \geq \frac{p_* \psi}{4} \mid \mathcal{A} \right] \geq 1 - \left(d \exp \left(-\frac{p_* \psi |\mathcal{B}_m^j|}{16 d x_{\max}^2} \right) + 2 \exp \left(-\frac{p_* |\mathcal{B}_m^j|}{10} \right) \right).$$

Further on the event \mathcal{M}_m^j and $\mathcal{M}_{\bar{m}}^j$ in (38), it holds that

$$\mathbb{P} \left[\lambda_{\min}(\widehat{\Sigma}(\mathcal{B}_{\bar{m},k}^j)) \geq \frac{p_* \psi}{24} \mid \mathcal{A} \right] \geq 1 - \left(d \exp \left(-\frac{p_* \psi |\mathcal{B}_m^j|}{16 d x_{\max}^2} \right) + 2 \exp \left(-\frac{p_* |\mathcal{B}_m^j|}{10} \right) \right).$$

Proof of Lemma 19 Define the following event analogous to (30)

$$\bar{\mathcal{E}}_{m,k}^j = \left\{ \lambda_{\min}(\widehat{\Sigma}(\bar{\mathcal{B}}_{m,k}^j)) \geq \frac{\psi}{2} \right\}.$$

By Lemma 10 and the fact that $|\mathcal{B}_m^j| \geq |\mathcal{B}_{m,k}^j|$, it holds that $\lambda_{\min}(\widehat{\Sigma}(\mathcal{B}_{m,k}^j)) \geq p_* \psi / 4$ on the events $\bar{\mathcal{D}}_{m,k}^j$ in (37) and $\bar{\mathcal{E}}_{m,k}^j$ above. Thus, we have

$$\mathbb{P} \left[\lambda_{\min}(\widehat{\Sigma}(\mathcal{B}_{m,k}^j)) \leq \frac{p_* \psi}{4} \mid \mathcal{A} \right] \leq \mathbb{P} \left[(\bar{\mathcal{D}}_{m,k}^j)^c \cup (\bar{\mathcal{E}}_{m,k}^j)^c \mid \mathcal{A} \right] \leq \mathbb{P} \left[(\bar{\mathcal{E}}_{m,k}^j)^c \mid \bar{\mathcal{D}}_{m,k}^j, \mathcal{A} \right] + \mathbb{P} \left[(\bar{\mathcal{D}}_{m,k}^j)^c \mid \mathcal{A} \right].$$

Our first result then follows by noting that the first term above has

$$\mathbb{P} \left[(\bar{\mathcal{E}}_{m,k}^j)^c \mid \bar{\mathcal{D}}_{m,k}^j, \mathcal{A} \right] \leq d \exp \left(-\frac{p_* \psi |\mathcal{B}_m^j|}{16 d x_{\max}^2} \right),$$

corresponding to Lemma 9 given Lemma 16, and

$$\mathbb{P} \left[(\bar{\mathcal{D}}_{m,k}^j)^c \mid \mathcal{A} \right] \leq 2 \exp \left(-\frac{p_* |\mathcal{B}_m^j|}{10} \right),$$

according to Lemma 17.

Note that since $|\mathcal{B}_m^j| \geq |\mathcal{B}_{m,k}^j|$, the above also holds for $\lambda_{\min}(\widehat{\Sigma}(\mathcal{B}_m^j))$. Then, note that on the event \mathcal{M}_m^j and $\mathcal{M}_{\bar{m}}^j$, we have

$$\lambda_{\min}(\widehat{\Sigma}(\mathcal{B}_{\bar{m},k}^j)) \geq \frac{|\mathcal{B}_m^j|}{|\mathcal{B}_{\bar{m},k}^j|} \lambda_{\min}(\widehat{\Sigma}(\mathcal{B}_m^j)) \geq \frac{|\mathcal{B}_m^j|}{|\mathcal{B}_{\bar{m},k}^j|} \frac{p_* \psi}{4} \geq \frac{p_* \psi}{24},$$

where the last inequality uses Lemma 10 and the fact that $|\mathcal{B}_{\bar{m}}| = 2|\mathcal{B}_m|$. Our second result then follows. \square

Next, we show that Lemma 1 also holds for an adapted sequence of observations so that we can use all samples collected at arm k and instance j to debias in Step 2 for our all-sample estimators. We use the same notations as in Lemma 1 but use (\mathbf{X}^j, Y^j) to represent all the data collected at arm k and instance j through Algorithm 2 up to batch m , i.e., $\{(X_t, Y_t)\}_{t \in \mathcal{B}_{m,k}^j}$, in the following lemma.

LEMMA 20. *Define the following event for an adapted sequence of observations in $\mathcal{B}_{m,k}^j$*

$$\mathcal{H}^j(\mathcal{B}_{\bar{m},k}^j) = \left\{ \frac{2}{|\mathcal{B}_{\bar{m},k}^j|} \|\mathbf{X}^{j\top} \epsilon^j\|_{\infty} \leq \frac{\lambda_j}{2} \right\}. \quad (39)$$

Then, we have

$$\mathbb{P} \left[\mathcal{H}^j(\mathcal{B}_{\bar{m},k}^j) \right] \geq 1 - 2d \exp \left(-\frac{\lambda_j^2 |\mathcal{B}_{\bar{m},k}^j|}{32 \sigma_j^2 x_{\max}^2} \right).$$

Proof of Lemma 20 The proof is analogous to that of Lemma 1, using a concentration inequality for martingale sequence as in Lemma EC.2 in Bastani and Bayati (2020). \square

Now we provide the proof of Proposition 11. We follow a similar proof strategy as for Proposition 10 but now conditioned on \mathcal{A} . In addition, given arm k , we consider learning across a subset of instances of which arm k is an optimal arm, i.e., $\mathcal{W}_k \subseteq [N]$ defined in Assumption 9. This is because the suboptimal arms won't observe any users on the event of \mathcal{A} . We apply Proposition 5 to our all-sample estimators across the subset of instances in \mathcal{W}_k .

Proof of Proposition 11 Based on Lemma 6 and 16, our all-sample OLS estimators are subgaussian and independent across instances conditioned on $\{\mathcal{A}, \mathcal{F}_{m-1}\}$, and the conditions of Proposition 5 are satisfied. Yet, we make a small adjustment to Proposition 5, since we use batch data from $\{\mathcal{B}_{m,k}^i\}_{i \in [N]}$ to compute our trimmed mean estimator but all the data from $\mathcal{B}_{m,k}^j$ to debias for instance j in Step 2. In particular, we bound the event $\mathcal{H}^j(\mathcal{B}_{m,k}^j)$ in (39) (instead of $\mathcal{H}^j(\mathcal{B}_{m,k}^i)$), and additionally bound the event $\{\lambda_{\min}(\widehat{\Sigma}(\mathcal{B}_{m,k}^j)) \geq p_*\psi/24\}$ using Lemma 20 and 19 respectively. Then, applying Proposition 5, we can write

$$\begin{aligned} \mathbb{P} \left[\|\widehat{\beta}_{k,\bar{m}}^j - \beta_k^j\|_1 \geq \frac{144\lambda_j s}{p_*\zeta\psi} + 2C_0 d(3\zeta + 4\eta) \max_{i \in \mathcal{W}_k} \sqrt{\frac{\sigma_i^2}{p_*\psi|\mathcal{B}_{m,k}^i|}} \log\left(\frac{3}{\eta}\right) \middle| \mathcal{A}, \mathcal{F}_{m-1} \right] \\ \leq 3d \exp\left(-\frac{\rho N \eta^2}{9}\right) + \mathbb{P} \left[(\mathcal{H}^j(\mathcal{B}_{m,k}^j))^c \middle| \mathcal{A}, \mathcal{F}_{m-1} \right] + \mathbb{P} \left[\lambda_{\min}(\widehat{\Sigma}(\mathcal{B}_{m,k}^j)) \leq \frac{p_*\psi}{24} \middle| \mathcal{A}, \mathcal{F}_{m-1} \right] \\ + \sum_{i \in \mathcal{W}_k} \mathbb{P} \left[\lambda_{\min}(\widehat{\Sigma}(\mathcal{B}_{m,k}^i)) \leq \frac{p_*\psi}{4} \middle| \mathcal{A}, \mathcal{F}_{m-1} \right], \end{aligned}$$

Taking expectation over \mathcal{F}_{m-1} on both sides, we can obtain

$$\begin{aligned} \mathbb{P} \left[\|\widehat{\beta}_{k,\bar{m}}^j - \beta_k^j\|_1 \geq \frac{144\lambda_j s}{p_*\zeta\psi} + 2C_0 d(3\zeta + 4\eta) \max_{i \in \mathcal{W}_k} \sqrt{\frac{\sigma_i^2}{p_*\psi|\mathcal{B}_{m,k}^i|}} \log\left(\frac{3}{\eta}\right) \middle| \mathcal{A} \right] \\ \leq 3d \exp\left(-\frac{\rho N \eta^2}{9}\right) + 2d \exp\left(-\frac{\lambda_j^2 |\mathcal{B}_{m,k}^j|}{32\sigma_j^2 x_{\max}^2}\right) + \mathbb{P} \left[\lambda_{\min}(\widehat{\Sigma}(\mathcal{B}_{m,k}^j)) \leq \frac{p_*\psi}{24} \middle| \mathcal{A} \right] \\ + \sum_{i \in \mathcal{W}_k} \mathbb{P} \left[\lambda_{\min}(\widehat{\Sigma}(\mathcal{B}_{m,k}^i)) \leq \frac{p_*\psi}{4} \middle| \mathcal{A} \right], \end{aligned}$$

where we apply Lemma 20 and use $|\mathcal{B}_{m,k}^j| \geq |\mathcal{B}_{m,k}^i|$. In addition, on the events $\bar{\mathcal{D}}_{m,k}^j$ in (37), we have $|\mathcal{B}_{m,k}^j| \geq |\bar{\mathcal{B}}_{m,k}^j| \geq p_* |\mathcal{B}_m^j|/2$ by Lemma 16. Thus, with a union bound over $\bigcap_{i \in \mathcal{W}_k} \bar{\mathcal{D}}_{m,k}^i$, we have

$$\begin{aligned} \mathbb{P} \left[\|\widehat{\beta}_{k,\bar{m}}^j - \beta_k^j\|_1 \geq \frac{144\lambda_j s}{p_*\zeta\psi} + 2C_0 d(3\zeta + 4\eta) \max_{i \in \mathcal{W}_k} \sqrt{\frac{2\sigma_i^2}{p_*^2\psi|\mathcal{B}_m^i|}} \log\left(\frac{3}{\eta}\right) \middle| \mathcal{A} \right] \\ \leq 3d \exp\left(-\frac{\rho N \eta^2}{9}\right) + 2d \exp\left(-\frac{\lambda_j^2 p_* |\mathcal{B}_m^j|}{64\sigma_j^2 x_{\max}^2}\right) + \mathbb{P} \left[\lambda_{\min}(\widehat{\Sigma}(\mathcal{B}_{m,k}^j)) \leq \frac{p_*\psi}{24} \middle| \mathcal{A} \right] \\ + \sum_{i \in \mathcal{W}_k} \mathbb{P} \left[\lambda_{\min}(\widehat{\Sigma}(\mathcal{B}_{m,k}^i)) \leq \frac{p_*\psi}{4} \middle| \mathcal{A} \right] + \sum_{i \in \mathcal{W}_k} 2 \exp\left(-\frac{p_* |\mathcal{B}_m^i|}{10}\right), \quad (40) \end{aligned}$$

where we use Lemma 17. Similarly, we take $\zeta = \frac{C_0-2}{4C_0} \sqrt{\frac{s}{d}}$ and set

$$\eta = \sqrt{\frac{9 \log(d \min_{i \in \mathcal{W}_k} |\mathcal{B}_m^i|)}{\rho N}}, \quad \lambda_j = \sqrt{\frac{384\sigma_j^2 x_{\max}^2 \log(d |\mathcal{B}_m^j|)}{p_* |\mathcal{B}_m^j|}}.$$

Since $|\mathcal{B}_m^i| = 0$ for any $i \in [N] \setminus \mathcal{W}_k$ conditioned on the event \mathcal{A} , the value of η is equivalent to

$$\eta = \sqrt{\frac{9 \log(d \min_{i \in [N], |\mathcal{B}_m^i| > 0} |\mathcal{B}_m^i|)}{\rho N}}.$$

As $\log(d \min_{i \in [N], |\mathcal{B}_m^i| > 0} |\mathcal{B}_m^i|) \geq \log(d) \geq 1$, it holds that $\sqrt{\log(\frac{3}{\eta})} \leq \sqrt{\frac{\log(\rho N)}{2}}$. Then, using a union bound over $\bigcap_{i \in \mathcal{W}_k} (\mathcal{M}_m^i \cap \mathcal{M}_{\bar{m}}^i)$ and applying Lemma 19 and 18, we obtain from inequality (40) that

$$\begin{aligned} \mathbb{P} \left[\|\widehat{\beta}_{k, \bar{m}}^j - \beta_k^j\|_1 \geq C_1 \sqrt{\frac{sd \log(dp_j |\mathcal{B}_{\bar{m}}|/2)}{p_j |\mathcal{B}_{\bar{m}}|}} + C_2 \sqrt{\frac{sd \log(\rho N)}{p_j |\mathcal{B}_m|}} + C_3 d \sqrt{\frac{\log(dp_j |\mathcal{B}_m|/2) \log(\rho N)}{N p_j |\mathcal{B}_m|}} \middle| \mathcal{A} \right] \\ \leq \frac{6}{\min_{i \in \mathcal{W}_k} p_i |\mathcal{B}_m|} + \frac{8}{p_j |\mathcal{B}_{\bar{m}}|} + \sum_{i \in \mathcal{W}_k} 2d \exp \left(-\frac{p_* p_i \psi |\mathcal{B}_m|}{32 d x_{\max}^2} \right) + \sum_{i \in \mathcal{W}_k} 10 \exp \left(-\frac{p_* p_i |\mathcal{B}_m|}{20} \right) \\ \leq \frac{6}{\min_{i \in \mathcal{W}_k} p_i |\mathcal{B}_m|} + \frac{4}{p_j |\mathcal{B}_m|} + \sum_{i \in \mathcal{W}_k} 7d \exp \left(-\frac{p_* p_i \psi |\mathcal{B}_m|}{32 d x_{\max}^2} \right), \end{aligned}$$

where C_1 , C_2 , and C_3 are constants listed at the beginning of §C.3, we use $|\mathcal{B}_m|/|\mathcal{B}_{\bar{m}}| = 1/2$ for $m \geq 1$, the first inequality uses $\mathbb{P}[(\mathcal{M}_m^i)^c \cup (\mathcal{M}_{\bar{m}}^i)^c] \leq 4 \exp(-\frac{p_* p_i |\mathcal{B}_m|}{20})$, and the last inequality uses $\psi \leq d x_{\max}^2$. This gives us our final result.

In addition, to satisfy $\eta \leq 1/2 - 1/C_0 - \zeta$, we require

$$\log(d |\mathcal{B}_m|) \leq \left(\frac{C_0 - 2}{2C_0} \left(1 - \frac{1}{2} \sqrt{\frac{s}{d}} \right) \right)^2 \frac{\rho N}{9}.$$

Since $|\mathcal{B}_m| \leq T$, we conclude that it suffices to have $N = \Omega(\log(d) \log(T))$, combined with (36). \square

C.4. Regret Bound

Finally, we prove Theorem 2, which provides an upper bound on the cumulative regret across all bandit instances, by bounding the regret of the three cases in Appendix C.1. We start by providing a worst-case regret bound for Case (i).

LEMMA 21. *The cumulative expected regret from the first two batches, i.e., $\mathcal{B}_0 \cup \mathcal{B}_1$, or when $T \leq N$ is at most*

$$2bx_{\max}(2q \log(T) + N).$$

Proof of Lemma 21 By our design, we have $2q \log(T)$ time steps in total from $\mathcal{B}_0 \cup \mathcal{B}_1$. The worst-case regret per time step is $2bx_{\max}$. The result then follows. \square

Next, we discuss the worst-case regret bound for Case (ii).

LEMMA 22. *When \mathcal{A} does not hold but $T \geq N$, the cumulative expected regret from all the batches $\{\mathcal{B}_m\}_{m>1}$ up to time T is at most*

$$16bx_{\max}KN.$$

Proof of Lemma 22 By Lemma 14, the failure probability of \mathcal{A} is at most $8KN/T$. The worst-case cumulative regret is at most $2bx_{\max}T$ throughout $\{\mathcal{B}_m\}_{m>1}$. The result then follows. \square

Before we proceed to the regret analysis of Case (iii), we state the following lemma, which shows a sufficiently large amount of data is employed to train the all-sample estimators.

LEMMA 23. For any $t \in \mathcal{B}_m$ with $m > 1$, we have

$$|\mathcal{B}_{m-1}| \geq \frac{t}{4}, \quad |\mathcal{B}_{m-1}^-| \geq \frac{t}{2}.$$

Proof of Lemma 23 By our design, $|\mathcal{B}_m| = 2^{m-1}|\mathcal{B}_0|$ for any $m \geq 1$, which implies for any $m > 1$

$$\frac{|\mathcal{B}_{m-1}|}{t} \geq \frac{|\mathcal{B}_{m-1}|}{\sum_{i=0}^m |\mathcal{B}_i|} = \frac{1}{4}, \quad \frac{|\mathcal{B}_{m-1}^-|}{t} \geq \frac{\sum_{i=0}^{m-1} |\mathcal{B}_i|}{\sum_{i=0}^m |\mathcal{B}_i|} = \frac{1}{2}. \quad \square$$

The next lemma shows that the forced-sample estimators filter out suboptimal arms in Algorithm 2.

LEMMA 24. When \mathcal{A} holds, the set of arms \mathcal{K} that survive after using the forced-sample estimators contains the optimal arm $k = \arg \max_{i \in [K]} X_t^\top \beta_i^j$ given $Z_t = j$ and no suboptimal arms in \mathcal{K}_{sub}^j .

Proof of Lemma 24 The proof is similar to Lemma EC.18 in Bastani and Bayati (2020). \square

Next, we provide a per-period regret bound when a user is observed at time t and instance j in Case (iii).

LEMMA 25. When \mathcal{A} holds, $N = \Omega(\log(d) \log(T))$ and $Z_t = j$, the expected regret at time $t \in \mathcal{B}_m$ for $m > 1$ is upper bounded by

$$\begin{aligned} r_t^j \leq & 24x_{\max}^2 LK \left(C_1^2 \frac{sd \log(dp_j t)}{p_j t} + 2C_2^2 \frac{sd \log(\rho N)}{p_j t} + 2C_3^2 \frac{d^2 \log(\rho N) \log(dp_j t)}{N p_j t} \right) \\ & + 4bx_{\max} K \left(\max_{i \in [N]} \frac{24}{p_i t} + \frac{16}{p_j t} + 7dN \exp \left(-\frac{p_* \psi(\min_{i \in [N]} p_i) t}{128dx_{\max}^2} \right) \right). \end{aligned}$$

Proof of Lemma 25 Without loss of generality, assume arm 1 is optimal for X_t , i.e., $\arg \max_{k \in [K]} X_t^\top \beta_k^j = 1$. Note that here the optimal arm is a function of X_t and hence a random variable; for simplicity, we fix arm 1 as the optimal arm in the following. Consider the conditional expected regret at time $t \in \mathcal{B}_m$

$$r_t^j(X_t) = \mathbb{E} \left[\sum_{k \in \mathcal{K}} X_t^\top (\beta_1^j - \beta_k^j) \mathbb{1}(\pi_{m-1}^j(X_t) = k) \middle| X_t, Z_t = j, \mathcal{A} \right],$$

where the set of arms \mathcal{K} is defined in Algorithm 2. Since $\pi_{m-1}^j(X_t) = k$ implies $X_t^\top \hat{\beta}_{k,m-1}^j \geq X_t^\top \hat{\beta}_{1,m-1}^j$, we have

$$r_t^j(X_t) \leq \mathbb{E} \left[\sum_{k \in \mathcal{K}} X_t^\top (\beta_1^j - \beta_k^j) \mathbb{1}(X_t^\top \hat{\beta}_{k,m-1}^j \geq X_t^\top \hat{\beta}_{1,m-1}^j) \middle| X_t, Z_t = j, \mathcal{A} \right]. \quad (41)$$

Define the event

$$\mathcal{L}_k^j = \{2x_{\max} \delta \leq X_t^\top (\beta_1^j - \beta_k^j)\}.$$

Then, we can decompose the upper bound of the regret in (41) into two parts given \mathcal{L}_k^j , that is,

$$r_t^j(X_t) \leq \sum_{r=1,2} r_{t,r}^j(X_t), \quad (42)$$

where

$$r_{t,1}^j(X_t) = \mathbb{E} \left[\sum_{k \in \mathcal{K}} X_t^\top (\beta_1^j - \beta_k^j) \mathbb{1}(\{X_t^\top \hat{\beta}_{k,m-1}^j \geq X_t^\top \hat{\beta}_{1,m-1}^j\} \cap \mathcal{L}_k^j) \middle| X_t, Z_t = j, \mathcal{A} \right], \quad (43)$$

$$r_{t,2}^j(X_t) = \mathbb{E} \left[\sum_{k \in \mathcal{K}} X_t^\top (\beta_1^j - \beta_k^j) \mathbb{1}(\{X_t^\top \hat{\beta}_{k,m-1}^j \geq X_t^\top \hat{\beta}_{1,m-1}^j\} \cap (\mathcal{L}_k^j)^c) \middle| X_t, Z_t = j, \mathcal{A} \right]. \quad (44)$$

On one hand, the event $\left\{X_t^\top \hat{\beta}_{k,m-1}^j \geq X_t^\top \hat{\beta}_{1,m-1}^j\right\} \cap \mathcal{L}_k^j$ regarding $r_{t,1}^j(X_t)$ implies

$$X_t^\top (\hat{\beta}_{k,m-1}^j - \beta_k^j) - X_t^\top (\hat{\beta}_{1,m-1}^j - \beta_1^j) \geq X_t^\top (\beta_1^j - \beta_k^j) \geq 2x_{\max} \delta.$$

Thus, at least one of $|X_t^\top (\hat{\beta}_{\iota,m-1}^j - \beta_\iota^j)|$ with $\iota \in \{1, k\}$ must be greater than $x_{\max} \delta$, which means

$$\begin{aligned} & \mathbb{E} \left[\mathbb{1} \left(\{X_t^\top \hat{\beta}_{k,m-1}^j \geq X_t^\top \hat{\beta}_{1,m-1}^j\} \cap \mathcal{L}_k^j \right) \mid X_t, Z_t = j, \mathcal{A} \right] \\ & \leq \sum_{\iota \in \{1, k\}} \mathbb{P} \left[|X_t^\top (\hat{\beta}_{\iota,m-1}^j - \beta_\iota^j)| \geq x_{\max} \delta \mid X_t, Z_t = j, \mathcal{A} \right] \leq \sum_{\iota \in \{1, k\}} \mathbb{P} \left[\|\hat{\beta}_{\iota,m-1}^j - \beta_\iota^j\|_1 \geq \delta \mid \mathcal{A} \right]. \end{aligned}$$

Since Lemma 24 implies both arm 1 and k are not suboptimal, we further upper bound the above probability using our all-sample tail inequality, i.e., Proposition 11. Together with Lemma 23, we can write

$$\mathbb{P} \left[\|\hat{\beta}_{\iota,m-1}^j - \beta_\iota^j\|_1 \geq \delta \right] \leq \max_{i \in [N]} \frac{24}{p_i t} + \frac{16}{p_j t} + 7dN \exp \left(-\frac{p_* \psi(\min_{i \in [N]} p_i) t}{128dx_{\max}^2} \right),$$

for $\iota \in \{1, k\}$, where

$$\delta = C_1 \sqrt{\frac{2sd \log(dp_j t)}{p_j t}} + C_2 \sqrt{\frac{4sd \log(\rho N)}{p_j t}} + C_3 d \sqrt{\frac{4 \log(dp_j t) \log(\rho N)}{N p_j t}}.$$

Then, we can obtain from (43) that

$$\begin{aligned} \mathbb{E} [r_{t,1}^j(X_t) \mid Z_t = j, \mathcal{A}] & \leq 2bx_{\max} K \sum_{\iota \in \{1, k\}} \mathbb{P} \left[\|\hat{\beta}_{\iota,m-1}^j - \beta_\iota^j\|_1 \geq \delta \right] \\ & \leq 4bx_{\max} K \left(\max_{i \in [N]} \frac{24}{p_i t} + \frac{16}{p_j t} + 7dN \exp \left(-\frac{p_* \psi(\min_{i \in [N]} p_i) t}{128dx_{\max}^2} \right) \right). \end{aligned} \quad (45)$$

On the other hand, by Assumption 8, we have for the term $r_{t,2}^j(X_t)$ that

$$\mathbb{E} [r_{t,2}^j(X_t) \mid Z_t = j, \mathcal{A}] \leq 2x_{\max} \delta K \mathbb{P}[(\mathcal{L}_k^j)^c] \leq 4x_{\max}^2 LK \delta^2. \quad (46)$$

Combining (45) and (46) with (42), we can get our result by using the inequality that $3(a^2 + b^2 + c^2) \geq (a + b + c)^2$. \square

Given the results above, we can obtain a cumulative regret bound for all bandit instances for Case (iii).

LEMMA 26. *When \mathcal{A} holds, $N = \Omega(\log(d) \log(T))$ and $T \geq N$, the cumulative expected regret from the batches $\{\mathcal{B}_m\}_{m>1}$ up to time T is upper bounded by*

$$\begin{aligned} & \sum_{j \in [N]} \left[24x_{\max}^2 LK \left(C_1^2 s d \log(dp_j T) + 2C_2^2 s d \log(\rho N) + 2C_3^2 \frac{d^2 \log(\rho N) \log(dp_j T)}{N} \right) \log(p_j T) \right. \\ & \quad \left. + 4bx_{\max} K \left(\left(16 + \max_{i \in [N]} \frac{24p_j}{p_i} \right) \log(p_j T) + \max_{i \in [N]} \frac{896x_{\max}^2 d^2 p_j}{p_* \psi p_i N} \right) \right]. \end{aligned}$$

Proof of Lemma 26 The cumulative expected regret from all instances over $\{\mathcal{B}_m\}_{m>1}$ is

$$\mathbb{E} \left[\sum_{t \in \bigcup_{m>1} \mathcal{B}_m} r_t^{Z_t}(X_t) \mid \mathcal{A} \right] = \sum_{t \in \bigcup_{m>1} \mathcal{B}_m} \mathbb{E} [\mathbb{E} [r_t^{Z_t}(X_t) \mid Z_t, \mathcal{A}] \mid \mathcal{A}] = \sum_{t=2q \log(T)+1}^T \sum_{j \in [N]} p_j r_t^j. \quad (47)$$

Note that we have

$$\int_{2q \log(T)}^T \frac{1}{p_j t} dt \leq \frac{\log(p_j T)}{p_j}.$$

Moreover, given q in Proposition 10, we have

$$\begin{aligned} \int_{2q \log(T)}^T dN \exp\left(-\frac{p_* \psi(\min_{i \in [N]} p_i) t}{128 d x_{\max}^2}\right) dt &\leq \frac{128 d^2 N x_{\max}^2}{p_* \psi(\min_{i \in [N]} p_i)} \exp\left(-\frac{p_* \psi(\min_{i \in [N]} p_i) q \log(T)}{64 d x_{\max}^2}\right) \\ &\leq \frac{128 d^2 N x_{\max}^2}{p_* \psi(\min_{i \in [N]} p_i) T^{3K \log(dN)/2}} \\ &\leq \frac{128 x_{\max}^2 d^2}{p_* \psi(\min_{i \in [N]} p_i) N}, \end{aligned}$$

where the last inequality holds since $T^{3K \log(dN)/2} \geq N^2$ when $T \geq N$ and $K, d, N > 1$. Combined all the above with Lemma 25, the result then follows. \square

Now we are ready to prove Theorem 2.

Proof of Theorem 2 Summing up the cumulative expected regrets of the three cases obtained in Lemma 21, 22 and 26, the cumulative expected regret across N instances up to time T satisfies

$$\begin{aligned} R_T &\leq 2b x_{\max} (2q \log(T) + N) + 16b x_{\max} K N \\ &\quad + \sum_{j \in [N]} \left[24 x_{\max}^2 L K \left(C_1^2 s d \log(dp_j T) + 2C_2^2 s d \log(\rho N) + 2C_3^2 \frac{d^2 \log(\rho N) \log(dp_j T)}{N} \right) \log(p_j T) \right. \\ &\quad \left. + 4b x_{\max} K \left((16 + \max_{i \in [N]} \frac{24 p_j}{p_i}) \log(p_j T) + \max_{i \in [N]} \frac{896 x_{\max}^2 d^2 p_j}{p_* \psi p_i N} \right) \right]. \end{aligned}$$

In the standard case, $p_i = \Theta(1/N)$ for any $i \in [N]$ and thereby $q = \mathcal{O}(Kd(sN + d) \log(d) \log(N))$ in Proposition 10. The claim then follows. \square

C.5. Margin Condition

In this section, we discuss how our regret bound in Theorem 2 is affected by the margin condition (Assumption 8). We assume a more general margin condition (Bastani et al. 2021a) as follows, and show that our algorithm can still achieve an improvement regarding the context dimension.

ASSUMPTION 10 (α -Margin Condition). *For any arms k and k' of any instance $j \in [N]$, there exists a constant $L > 0$ such that $\mathbb{P}[|X^\top(\beta_k^j - \beta_{k'}^j)| \leq \kappa \mid Z = j] \leq L\kappa^\alpha$ for any $\kappa > 0$ and for some $\alpha \geq 0$.*

It's easy to find that throughout our regret analysis in Appendix C.4, we only use the margin condition in Lemma 25. Therefore, given Assumption 10, we have the following analog of Lemma 25:

LEMMA 27. *When \mathcal{A} holds, $N = \Omega(\log(d) \log(T))$, $Z_t = j$ and $\alpha \neq 1$, the expected regret at time $t \in \mathcal{B}_m$ for $m > 1$ is upper bounded by*

$$\begin{aligned} r_t^j &\leq 24 x_{\max}^2 L K \left(C_1 \sqrt{s d \log(dp_j T)} + C_2 \sqrt{2 s d \log(\rho N)} + C_3 \sqrt{\frac{2 d^2 \log(\rho N) \log(dp_j T)}{N}} \right)^{\alpha+1} (p_j t)^{-\frac{\alpha+1}{2}} \\ &\quad + 4b x_{\max} K \left(\max_{i \in [N]} \frac{24}{p_i t} + \frac{16}{p_j t} + 7dN \exp\left(-\frac{p_* \psi(\min_{i \in [N]} p_i) t}{128 d x_{\max}^2}\right) \right). \end{aligned}$$

Proof of Lemma 27 The proof is the same as Lemma 25. We follow the same steps until (45). Now instead using Assumption 10, we have for the term $r_{t,2}^j(X_t)$ that

$$\mathbb{E}[r_{t,2}^j(X_t) \mid Z_t = j, \mathcal{A}] \leq 2x_{\max} \delta K \mathbb{P}[(\mathcal{L}_k^j)^c] \leq 4x_{\max}^2 L K \delta^{\alpha+1}. \quad (48)$$

The claim then follows. \square

Now we are ready to state the following regret bound for all instances given our general margin condition. Intuitively, a larger value of α imposes stronger assumptions on the contextual distribution \mathcal{P}_X^j around the boundary — i.e., it rules out distributions with high density around the boundary — and hence makes the problem easier to learn. When $\alpha \rightarrow 0$, we obtain a $\mathcal{O}(\sqrt{T})$ regret guarantee; however, when $\alpha \rightarrow 1$, we recover an optimal $\mathcal{O}(\log(T))$ regret as stated in Theorem 2. The proof is similar to that of Theorem 2.

PROPOSITION 12. *When $N = \Omega(\log(d)\log(T))$, the total cumulative expected regret of all instances up to time T implementing Algorithm 2 satisfies*

$$R_T = \mathcal{O} \left(KN \left(sd + \frac{d^2}{N} \right)^{\frac{\alpha+1}{2}} (\log(N) \log(dT/N))^{\frac{\alpha+1}{2}} (T/N)^{\frac{1-\alpha}{2}} \right)$$

when $\alpha < 1$ and

$$R_T = \mathcal{O}(KN \log(T/N))$$

when $\alpha > 1$, for appropriate choices of hyperparameters $\omega_0, \zeta_{1,0}, \eta_{1,0}, \lambda_0, \lambda_{1,0}$, and q provided in Appendix C.

C.6. Standard Regime

The proof is a direct application of Theorem 2 by considering an expected time horizon of T for instance j .

Proof of Corollary 3 The cumulative expected regret of any instance j in Case (iii) is

$$\mathbb{E} \left[\sum_{t \in \bigcup_{m>1} \mathcal{B}_m} r_t^{Z_t}(X_t) \mathbb{1}(Z_t = j) \middle| \mathcal{A} \right] = \sum_{t \in \bigcup_{m>1} \mathcal{B}_m} p_j r_t^j.$$

Corresponding to the proof of Lemma 26, we have

$$\begin{aligned} \sum_{t \in \bigcup_{m>1} \mathcal{B}_m} p_j r_t^j &\leq 24x_{\max}^2 LK \left(C_1^2 sd \log(dT) + 2C_2^2 sd \log(\rho N) + 2C_3^2 \frac{d^2 \log(\rho N) \log(dT)}{N} \right) \log(T) \\ &\quad + 4bx_{\max} K \left((16 + \max_{i \in [N]} \frac{24p_j}{p_i}) \log(T) + \max_{i \in [N]} \frac{896x_{\max}^2 d^2 p_j}{p_* \psi p_i N} \right). \end{aligned}$$

Besides, the cumulative expected regret of instance j from Case (i) and (ii) is simply

$$p_j (2bx_{\max}(2q \log(T) + N) + 16bx_{\max}KN).$$

Combining all the above with $p_j = \Theta(1/N)$ in the standard case, we can get our result. \square

C.7. Data-Poor Regime

In this section, we give a proof and the hyperparameter choices for Corollary 4. The proof follows closely that of Theorem 2. We list the hyperparameter choices for Corollary 4 as follows:

$$\omega_0 = \zeta_0 + \eta_0, \quad \zeta_0 = \zeta_{1,0} = \frac{C_0 - 2}{4C_0}, \quad \eta_0 = \eta_{1,0} = 0, \quad \lambda_{0,j} = \frac{p_* \psi h}{256x_{\max} s}, \quad \lambda_{1,j,0} = \sqrt{\frac{64\sigma_j^2 x_{\max}^2}{p_*}},$$

and

$$q = \max \left\{ \frac{2 \cdot 384^3 C_0^2 x_{\max}^4 (\max\{\max_{i \neq j} \sigma_i^2 sd/p_i, \sigma_j^2 s^2/p_j\}) K \log(d)}{(C_0 - 2)^2 h^2 p_*^2 \psi^2}, \frac{576 C_0^2 x_{\max}^2 (\max_{i \neq j} d^2 \sigma_i^2/p_i) K s^2 \log(N)}{h^2 p_* \psi}, \right. \\ \left. \frac{6 \cdot 192^2 C_0^2 x_{\max}^2 (\max_{i \neq j} d^2 \sigma_i^2/p_i) K \log(d) \log(N)}{h^2 p_* \psi N}, \frac{96 x_{\max}^2 K d \log(dN)}{p_* \psi (\min_{i \neq j} p_i)}, \frac{20K}{p_* p_j} \right\}.$$

In the online learning setting, the data-poor instance j might receive far fewer observations than the time horizon; therefore, our following analysis assumes the compatibility condition (Assumption 4) for the data-poor instance j .

Forced-Sample Estimator. First, we provide an analog of Proposition 10 for a data-poor regime, which states that our forced-sample estimators have small estimation errors with high probability.

PROPOSITION 13. *When $N = \Omega(\log(d) \log(T))$, the forced-sample estimator $\widehat{\beta}_{k,0}^j = \widehat{\beta}_k^j(\mathcal{B}_0, \lambda_{0,j}, \omega_0)$ of data-poor instance j and arm k in Algorithm 2 satisfies*

$$\mathbb{P} \left[\|\widehat{\beta}_{k,0}^j - \beta_k^j\|_1 \geq \frac{h}{4x_{\max}} \right] \leq \frac{13}{T},$$

for the hyperparameter choices $\zeta_0, \eta_0, \lambda_0$, and q specified in Appendix C.7.

Before we prove Proposition 13, we first introduce several useful lemmas.

LEMMA 28. *When $|\bar{\mathcal{B}}_{0,k}^j| \geq 3 \log(d)/C'^2$, we have*

$$\mathbb{P} \left[\widehat{\Sigma}(\bar{\mathcal{B}}_{0,k}^j) \in \mathcal{C}(\bar{\mathcal{I}}_j, \psi/2) \right] \geq 1 - \exp(-C'^2 |\bar{\mathcal{B}}_{0,k}^j|),$$

where $C' = \max \left\{ \frac{1}{2}, \frac{\zeta \psi}{768 s x_{\max}^2} \right\}$.

Proof of Lemma 28 The proof is the same as Lemma 4. \square

LEMMA 29. *For any sets $\mathcal{B}, \bar{\mathcal{B}}$ with $\bar{\mathcal{B}} \subseteq \mathcal{B}$, if $\widehat{\Sigma}(\bar{\mathcal{B}}) \in \mathcal{C}(\mathcal{S}, \phi)$, then $\widehat{\Sigma}(\mathcal{B}) \in \mathcal{C}(\mathcal{S}, |\bar{\mathcal{B}}|\phi/|\mathcal{B}|)$.*

Proof of Lemma 29 See Lemma EC.7 in Bastani and Bayati (2020). \square

Now we are ready to prove Proposition 13 by using Corollary 5.

Proof of Proposition 13 The proof mainly follows that of Proposition 10. Applying Corollary 5 by setting $\phi = \tilde{\phi} = p_* \psi/4$, using Lemma 28 and 29, and using $|\mathcal{B}_{0,k}^j| = |\mathcal{B}_0^j|/K$, we have

$$\begin{aligned} \mathbb{P} \left[\|\widehat{\beta}_{k,0}^j - \beta_k^j\|_1 \geq \frac{24\lambda_j s}{p_* \zeta \psi} + 2C_0 d(3\zeta + 4\eta) \max_{i \neq j} \sqrt{\frac{K \sigma_i^2}{p_* \psi |\mathcal{B}_0^i|} \log\left(\frac{3}{\eta}\right)} \right] \\ \leq 3d \exp\left(-\frac{N\eta^2}{9}\right) + 2d \exp\left(-\frac{\lambda_j^2 |\mathcal{B}_{0,k}^j|}{32\sigma_j^2 x_{\max}^2}\right) + \mathbb{P} \left[\widehat{\Sigma}(\mathcal{B}_{0,k}^j) \in \mathcal{C}(\bar{\mathcal{I}}_j, \frac{p_* \psi}{4}) \right] + \sum_{i \neq j} \mathbb{P} \left[\lambda_{\min}(\widehat{\Sigma}(\mathcal{B}_{0,k}^i)) \leq \frac{p_* \psi}{4} \right] \\ \leq 3d \exp\left(-\frac{N\eta^2}{9}\right) + 2d \exp\left(-\frac{\lambda_j^2 |\mathcal{B}_0^j|}{32K \sigma_j^2 x_{\max}^2}\right) + \exp\left(-\frac{p_* C'^2 |\mathcal{B}_0^j|}{2K}\right) \\ + \sum_{i \neq j} d \exp\left(-\frac{p_* \psi |\mathcal{B}_0^i|}{16K d x_{\max}^2}\right) + \sum_{i \in [N]} 2 \exp\left(-\frac{p_* |\mathcal{B}_0^i|}{10K}\right), \end{aligned}$$

Correspondingly, take (recall that $|\mathcal{B}_0| = q \log(T)$)

$$\zeta = \frac{C_0 - 2}{4C_0}, \quad \eta = \sqrt{\frac{27 \log(d) |\mathcal{B}_0|}{qN}}, \quad \lambda_j = \sqrt{\frac{96 \sigma_j^2 x_{\max}^2 K \log(d) |\mathcal{B}_0|}{q |\mathcal{B}_0^j|}}.$$

With a union bound over $\bigcap_{i \in [N]} \mathcal{M}_0^i$ in (32) by Lemma 12, we have

$$\begin{aligned} \mathbb{P} \left[\|\widehat{\beta}_{k,0}^j - \beta_k^j\|_1 \geq \frac{768C_0 \sigma_j x_{\max} s}{(C_0 - 2) p_* \psi} \sqrt{\frac{3K \log(d)}{q p_j}} + C_0 \left(\frac{3s}{2} + 24 \sqrt{\frac{3 \log(d) |\mathcal{B}_0|}{qN}} \right) \max_{i \neq j} \sqrt{\frac{K d^2 \sigma_i^2 \log(N)}{2 p_* \psi p_i |\mathcal{B}_0|}} \right] \\ \leq \frac{5}{T} + \exp\left(-\frac{p_* p_j C'^2 |\mathcal{B}_0|}{4K}\right) + \sum_{i \neq j} d \exp\left(-\frac{p_* \psi p_i |\mathcal{B}_0|}{32K d x_{\max}^2}\right) + \sum_{i \in [N]} 4 \exp\left(-\frac{p_* p_i |\mathcal{B}_0|}{20K}\right) \\ \leq \frac{5}{T} + 5 \exp\left(-\frac{p_* p_j |\mathcal{B}_0|}{20K}\right) + \sum_{i \neq j} 3d \exp\left(-\frac{p_* \psi p_i |\mathcal{B}_0|}{32K d x_{\max}^2}\right), \end{aligned}$$

where the last inequality uses $C' \geq 1/2$. Next, we choose a sufficiently large q such that

$$\begin{aligned} \frac{h}{8x_{\max}} &\geq \frac{768C_0\sigma_j x_{\max}s}{(C_0-2)p_*\psi} \sqrt{\frac{3K\log(d)}{qp_j}}, \quad \frac{h}{16x_{\max}} \geq C_0 \frac{3s}{2} \max_{i \neq j} \sqrt{\frac{Kd^2\sigma_i^2 \log(N)}{2p_*\psi p_i |\mathcal{B}_0|}}, \\ \frac{h}{16x_{\max}} &\geq 24C_0 \sqrt{\frac{3\log(d)|\mathcal{B}_0|}{qN}} \max_{i \neq j} \sqrt{\frac{Kd^2\sigma_i^2 \log(N)}{2p_*\psi p_i |\mathcal{B}_0|}}, \quad \frac{3}{T} \geq \sum_{i \neq j} 3d \exp\left(-\frac{p_*\psi p_i |\mathcal{B}_0|}{32Kdx_{\max}^2}\right), \\ \frac{5}{T} &\geq 5 \exp\left(-\frac{p_*p_j |\mathcal{B}_0|}{20K}\right), \end{aligned}$$

and it suffices to set

$$q = \max \left\{ \frac{2 \cdot 384^3 C_0^2 x_{\max}^4 (\max\{\max_{i \neq j} \sigma_i^2 s d / p_i, \sigma_j^2 s^2 / p_j\}) K \log(d)}{(C_0-2)^2 h^2 p_*^2 \psi^2}, \frac{576 C_0^2 x_{\max}^2 (\max_{i \neq j} d^2 \sigma_i^2 / p_i) K s^2 \log(N)}{h^2 p_* \psi}, \right. \\ \left. \frac{6 \cdot 192^2 C_0^2 x_{\max}^2 (\max_{i \neq j} d^2 \sigma_i^2 / p_i) K \log(d) \log(N)}{h^2 p_* \psi N}, \frac{96 x_{\max}^2 K d \log(dN)}{p_* \psi (\min_{i \neq j} p_i)}, \frac{20K}{p_* p_j} \right\}.$$

Additionally, we also require $\eta \leq 1/2 - 1/C_0 - \zeta$ according to Proposition 5, which is satisfied as long as

$$\log(T) \leq \left(\frac{C_0-2}{4C_0}\right)^2 \frac{N-1}{27\log(d)}. \quad \square$$

All-Sample Estimator. Next, we provide a tail inequality of our all-sample estimator for the data-poor instance. The constants stated in Proposition 14 are

$$C_4 = \frac{96^2 C_0 \sigma_j x_{\max}}{(C_0-2)p_*^{3/2}\psi}, \quad C_5 = \frac{3C_0}{2p_*} (\max_{i \neq j} \sqrt{\frac{2\sigma_i^2 d^2 p_j}{\psi p_i}}), \quad C_6 = \frac{24C_0}{p_*} (\max_{i \neq j} \sqrt{\frac{2\sigma_i^2 d^2 p_j}{\psi p_i}}),$$

and our hyperparameter choices for all-sample estimators are

$$\zeta_{1,m} = \zeta_{1,0}, \quad \eta_{1,m} = \eta_{1,0} \sqrt{\log(d \min_{i \neq j, |\mathcal{B}_m^i| > 0} |\mathcal{B}_m^i|)}, \quad \lambda_{1,j,\bar{m}} = \lambda_{1,0} \sqrt{\frac{\log(d|\mathcal{B}_{\bar{m}}^j|)}{|\mathcal{B}_{\bar{m}}^j|}}.$$

For simplicity, we use ζ , η and λ_j to represent $\zeta_{1,m}$, $\eta_{1,m}$ and $\lambda_{1,j,\bar{m}}$ respectively in the following.

PROPOSITION 14. When \mathcal{A} holds and $N = \Omega(\log(d)\log(T))$, the all-sample estimator $\hat{\beta}_{k,\bar{m}}^j = \hat{\beta}_k^j(\mathcal{B}_{\bar{m}}, \lambda_{1,j,\bar{m}}, \omega_{1,m})$ of data-poor instance j and optimal arm $k \in \mathcal{K}_{opt}^j$ in Algorithm 2 satisfies

$$\|\hat{\beta}_{k,\bar{m}}^j - \beta_k^j\|_1 \leq C_4 \sqrt{\frac{s^2 \log(dp_j |\mathcal{B}_{\bar{m}}|)}{p_j |\mathcal{B}_{\bar{m}}|}} + C_5 \sqrt{\frac{\log(\rho N)}{p_j |\mathcal{B}_{\bar{m}}|}} + C_6 \sqrt{\frac{\log(dp_j |\mathcal{B}_{\bar{m}}|) \log(\rho N)}{N p_j |\mathcal{B}_{\bar{m}}|}}$$

with probability at least $1 - \left(\frac{6}{\min_{i \in \mathcal{W}_k, i \neq j} p_i |\mathcal{B}_{\bar{m}}|} + \frac{4}{p_j |\mathcal{B}_{\bar{m}}|} + 9 \exp\left(-\frac{p_* p_j |\mathcal{B}_{\bar{m}}|}{20}\right) + \sum_{i \in \mathcal{W}_k, i \neq j} 5d \exp\left(-\frac{p_* p_i \psi |\mathcal{B}_{\bar{m}}|}{32dx_{\max}^2}\right)\right)$ for the hyperparameter choices $\zeta_{1,0}$, $\eta_{1,0}$, and $\lambda_{1,0}$ specified in Corollary 4, where the constants C_4 , C_5 and C_6 are provided in Appendix C.7.

LEMMA 30. The event \mathcal{A} holds with at least a probability of $1 - 13KN/T$.

Proof of Lemma 30 The proof is similar to that of Lemma 14. \square

LEMMA 31. On the event \mathcal{M}_m^j and $\mathcal{M}_{\bar{m}}^j$ in (38), when $|\bar{\mathcal{B}}_{m,k}^j| \geq 3\log(d)/C'^2$, we have

$$\mathbb{P}\left[\widehat{\Sigma}(\bar{\mathcal{B}}_{m,k}^j) \in \mathcal{C}(\bar{\mathcal{I}}_j, p_*\psi/24)\right] \geq 1 - \left(\exp\left(-\frac{C'^2 |\bar{\mathcal{B}}_{m,k}^j|}{2}\right) + 2 \exp\left(-\frac{p_* |\bar{\mathcal{B}}_{m,k}^j|}{10}\right)\right),$$

where $C' = \max\left\{\frac{1}{2}, \frac{\zeta\psi}{768sx_{\max}^2}\right\}$.

Proof of Lemma 31 The proof follows closely that of Lemma 19. We list the details as follows. Define the following event

$$\tilde{\mathcal{E}}_{m,k}^j = \left\{ \widehat{\Sigma}(\mathcal{B}_{m,k}^j) \in \mathcal{C}(\bar{\mathcal{I}}_j, \frac{\psi}{2}) \right\}.$$

By Lemma 29 and the fact that $|\mathcal{B}_m^j| \geq |\mathcal{B}_{m,k}^j|$, it holds that $\widehat{\Sigma}(\mathcal{B}_{m,k}^j) \in \mathcal{C}(\bar{\mathcal{I}}_j, p_*\psi/4)$ on the events $\bar{\mathcal{D}}_{m,k}^j$ in (37) and $\tilde{\mathcal{E}}_{m,k}^j$ above. Thus, we have

$$\mathbb{P} \left[\widehat{\Sigma}(\mathcal{B}_{m,k}^j) \notin \mathcal{C}(\bar{\mathcal{I}}_j, \frac{p_*\psi}{4}) \middle| \mathcal{A} \right] \leq \mathbb{P} \left[(\bar{\mathcal{D}}_{m,k}^j)^c \cup (\tilde{\mathcal{E}}_{m,k}^j)^c \middle| \mathcal{A} \right] \leq \exp \left(-\frac{C'^2 |\mathcal{B}_m^j|}{2} \right) + 2 \exp \left(-\frac{p_* |\mathcal{B}_m^j|}{10} \right),$$

where we use Lemma 28 in the last inequality.

Note that since $|\mathcal{B}_m^j| \geq |\mathcal{B}_{m,k}^j|$, the above also holds for $\widehat{\Sigma}(\mathcal{B}_m^j)$. In addition, on the event \mathcal{M}_m^j and $\mathcal{M}_{\bar{m}}^j$, we have $|\mathcal{B}_m^j|/|\mathcal{B}_{\bar{m}}^j| \geq 1/6$. Therefore, given $\widehat{\Sigma}(\mathcal{B}_{m,k}^j) \in \mathcal{C}(\bar{\mathcal{I}}_j, \frac{p_*\psi}{4})$, we have by Lemma 29

$$\widehat{\Sigma}(\mathcal{B}_{\bar{m},k}^j) \in \mathcal{C}(\bar{\mathcal{I}}_j, \frac{p_*\psi}{24}).$$

Our result then follows. \square

Now we are ready to prove Proposition 14.

Proof of Proposition 14 The proof is similar to that of Proposition 11. We list the details that differ from Proposition 11 as follows.

Applying Corollary 5, together with a union bound over $\bigcap_{i \in \mathcal{W}_k} \bar{\mathcal{D}}_{m,k}^i$ in (37), we have

$$\begin{aligned} \mathbb{P} \left[\|\widehat{\beta}_{k,\bar{m}}^j - \beta_k^j\|_1 \geq \frac{144\lambda_j s}{p_*\zeta\psi} + 2C_0(3\zeta + 4\eta) \max_{i \in \mathcal{W}_k, i \neq j} \sqrt{\frac{2d^2\sigma_i^2}{p_*^2\psi|\mathcal{B}_m^i|} \log\left(\frac{3}{\eta}\right)} \middle| \mathcal{A} \right] \\ \leq 3d \exp \left(-\frac{\rho N \eta^2}{9} \right) + 2d \exp \left(-\frac{\lambda_j^2 p_* |\mathcal{B}_m^j|}{64\sigma_j^2 x_{\max}^2} \right) + \mathbb{P} \left[\widehat{\Sigma}(\mathcal{B}_{\bar{m},k}^j) \notin \mathcal{C}(\bar{\mathcal{I}}_j, \frac{p_*\psi}{24}) \middle| \mathcal{A} \right] \\ + \sum_{i \in \mathcal{W}_k, i \neq j} \mathbb{P} \left[\lambda_{\min}(\widehat{\Sigma}(\mathcal{B}_{m,k}^i)) \leq \frac{p_*\psi}{4} \middle| \mathcal{A} \right] + \sum_{i \in \mathcal{W}_k} 2 \exp \left(-\frac{p_* |\mathcal{B}_m^i|}{10} \right), \end{aligned}$$

Take

$$\zeta = \frac{C_0 - 2}{4C_0}, \quad \eta = \sqrt{\frac{9 \log(d \min_{i \in \mathcal{W}_k, i \neq j} |\mathcal{B}_m^i|)}{\rho N}}, \quad \lambda_j = \sqrt{\frac{384\sigma_j^2 x_{\max}^2 \log(d|\mathcal{B}_m^j|)}{p_* |\mathcal{B}_m^j|}}.$$

Note that the value of η is equivalent to

$$\eta = \sqrt{\frac{9 \log(d \min_{i \neq j, |\mathcal{B}_m^i| > 0} |\mathcal{B}_m^i|)}{\rho N}},$$

since $|\mathcal{B}_m^i| = 0$ for any $i \in [N] \setminus \mathcal{W}_k$ conditioned on the event \mathcal{A} . Similarly, using a union bound over $\bigcap_{i \in \mathcal{W}_k} (\mathcal{M}_m^i \cap \mathcal{M}_{\bar{m}}^j)$ and applying Lemma 19, 31 and 18, we get

$$\begin{aligned} \mathbb{P} \left[\|\widehat{\beta}_{k,\bar{m}}^j - \beta_k^j\|_1 \geq C_4 \sqrt{\frac{s^2 \log(dp_j |\mathcal{B}_{\bar{m}}|/2)}{p_j |\mathcal{B}_{\bar{m}}|}} + C_5 \sqrt{\frac{\log(\rho N)}{p_j |\mathcal{B}_{\bar{m}}|}} + C_6 \sqrt{\frac{\log(dp_j |\mathcal{B}_m|/2) \log(\rho N)}{N p_j |\mathcal{B}_m|}} \middle| \mathcal{A} \right] \\ \leq \frac{6}{\min_{i \in \mathcal{W}_k, i \neq j} p_i |\mathcal{B}_m|} + \frac{8}{p_j |\mathcal{B}_{\bar{m}}|} + \exp \left(-\frac{C'^2 p_j |\mathcal{B}_m|}{4} \right) + \sum_{i \in \mathcal{W}_k, i \neq j} d \exp \left(-\frac{p_* p_i \psi |\mathcal{B}_m|}{32 d x_{\max}^2} \right) + \sum_{i \in \mathcal{W}_k} 8 \exp \left(-\frac{p_* p_i |\mathcal{B}_m|}{20} \right) \\ \leq \frac{6}{\min_{i \in \mathcal{W}_k, i \neq j} p_i |\mathcal{B}_m|} + \frac{4}{p_j |\mathcal{B}_{\bar{m}}|} + 9 \exp \left(-\frac{p_* p_j |\mathcal{B}_m|}{20} \right) + \sum_{i \in \mathcal{W}_k, i \neq j} 5d \exp \left(-\frac{p_* p_i \psi |\mathcal{B}_m|}{32 d x_{\max}^2} \right), \end{aligned}$$

where C_4 , C_5 , and C_6 are constants listed at the beginning of §C.3 and we use $C' \geq 1/2$. This gives us our final result.

In addition, to satisfy $\eta \leq 1/2 - 1/C_0 - \zeta$, we require

$$\log(d|\mathcal{B}_m|) \leq \left(\frac{C_0 - 2}{2C_0}\right)^2 \frac{\rho N}{9}. \quad \square$$

Regret Bound. For the regret analysis in data-poor regime, we consider the following three possible cases separately:

- (i). The forced-sample batch ($t \in \mathcal{B}_0$) or the first batch using all-sample estimator ($t \in \mathcal{B}_1$);
- (ii). All the remaining batches ($t \in \mathcal{B}_m$ for $m > 1$) such that \mathcal{A} does not hold;
- (iii). All the remaining batches ($t \in \mathcal{B}_m$ for $m > 1$) such that \mathcal{A} holds.

The regret of the first two cases can be bound similarly as in Appendix C.4. We provide an upper bound of the per-period regret of the data-poor instance j in Case (iii).

LEMMA 32. *When \mathcal{A} holds, $N = \Omega(\log(d)\log(T))$ and $Z_t = j$ for data-poor instance j , the expected regret at time $t \in \mathcal{B}_m$ for $m > 1$ is upper bounded by*

$$r_t^j \leq 24x_{\max}^2 LK \left(C_4^2 \frac{s^2 \log(dp_j t)}{p_j t} + 2C_5^2 \frac{\log(\rho N)}{p_j t} + 2C_6^2 \frac{\log(\rho N) \log(dp_j t)}{N p_j t} \right) + 4bx_{\max} K \left(\max_{i \neq j} \frac{24}{p_i t} + \frac{16}{p_j t} + 9 \exp\left(-\frac{p_* p_j t}{80}\right) + 5dN \exp\left(-\frac{p_* \psi(\min_{i \neq j} p_i) t}{128dx_{\max}^2}\right) \right).$$

Proof of Lemma 32 The proof follows closely that of Lemma 25. We list the details that differ from Lemma 25 as follows.

Applying Proposition 14 and Lemma 23, we can write

$$\mathbb{P} \left[\|\widehat{\beta}_{\iota, m-1}^j - \beta_{\iota}^j\|_1 \geq \delta \right] \leq \max_{i \neq j} \frac{24}{p_i t} + \frac{16}{p_j t} + 9 \exp\left(-\frac{p_* p_j t}{80}\right) + 5dN \exp\left(-\frac{p_* \psi(\min_{i \neq j} p_i) t}{128dx_{\max}^2}\right),$$

for $\iota \in \{1, k\}$, where

$$\delta = C_4 \sqrt{\frac{2s^2 \log(dp_j t)}{p_j t}} + C_5 \sqrt{\frac{4 \log(\rho N)}{p_j t}} + C_6 \sqrt{\frac{4 \log(dp_j t) \log(\rho N)}{N p_j t}}.$$

Then, we can obtain

$$\mathbb{E} [r_{t,1}^j(X_t) | Z_t = j, \mathcal{A}] \leq 4bx_{\max} K \left(\max_{i \neq j} \frac{24}{p_i t} + \frac{16}{p_j t} + 9 \exp\left(-\frac{p_* p_j t}{80}\right) + 5dN \exp\left(-\frac{p_* \psi(\min_{i \neq j} p_i) t}{128dx_{\max}^2}\right) \right).$$

The result then follows the rest of the proof of Lemma 25. \square

Now we are ready to prove Corollary 4.

Proof of Corollary 4 The proof follows closely that of Corollary 3. The cumulative expected regret of data-poor instance j in Case (iii) is

$$\mathbb{E} \left[\sum_{t \in \bigcup_{m>1} \mathcal{B}_m} r_t^{Z_t}(X_t) \mathbb{1}(Z_t = j) \middle| \mathcal{A} \right] = \sum_{t \in \bigcup_{m>1} \mathcal{B}_m} p_j r_t^j.$$

Corresponding to the proof of Lemma 26, we use Lemma 32 and get for Case (iii)

$$\sum_{t \in \bigcup_{m>1} \mathcal{B}_m} p_j r_t^j \leq 24x_{\max}^2 LK \left(C_4^2 s^2 \log(dT) + 2C_5^2 \log(\rho N) + 2C_6^2 \frac{\log(\rho N) \log(dT)}{N} \right) \log(T) \\ + 4bx_{\max} K \left(\left(16 + \max_{i \neq j} \frac{24p_j}{p_i} \right) \log(T) + \frac{720}{p_*} + \max_{i \in [N]} \frac{896x_{\max}^2 d^2 p_j}{p_* \psi p_i N} \right).$$

Besides, the cumulative expected regret of instance j in Case (i) and (ii) is simply

$$p_j (4bx_{\max} q \log(T) + 26bx_{\max} KN),$$

where we use Lemma 30. Combining all the above with $p_j/p_{j'} = \Theta(1/d^2)$ for any $j' \neq j$ in the data-poor case, the claim then follows. \square

Appendix D: Auxiliary Results

This appendix collects useful results from the literature.

LEMMA 33. *Let $X = [X_1 \cdots X_n]$ be a vector of n independent σ -subgaussian random variables with mean μ . Then, for any $a \in \mathbb{R}^n$ and $t \geq 0$, it holds that*

$$\mathbb{P}[|a^\top (X - \mu)| \geq t] \leq 2 \exp \left(-\frac{t^2}{2\sigma^2 \|a\|_2^2} \right).$$

Proof of Lemma 33 See Corollary 1.7 of Rigollet and Hütter (2015).

LEMMA 34. *Consider a sequence of independent random symmetric matrices $X_k \in \mathbb{R}^{d \times d}$, $k \in [n]$ with $\lambda_{\min}(X_k) \geq 0$ and $\lambda_{\max}(X_k) \leq L$ for any k . Let $\mu = \lambda_{\min}(\mathbb{E}[\sum_{k \in [n]} X_k])$. We have for $0 < t < 1$ that*

$$\mathbb{P} \left[\lambda_{\min} \left(\sum_{k \in [n]} X_k \right) \geq t\mu \right] \geq 1 - d \exp \left(-\frac{(1-t)^2 \mu}{2L} \right).$$

Proof of Lemma 34 See page 61 in Tropp (2015).

LEMMA 35. *Suppose X_1, \dots, X_n are n independent Bernoulli random variables with mean p_1, \dots, p_n respectively. Let $\mu = \sum_{i \in [n]} p_i$. Then, we have*

$$\mathbb{P} \left[\left| \sum_{i \in [n]} X_i - \mu \right| \geq \frac{\mu}{2} \right] \leq 2 \exp \left(-\frac{\mu}{10} \right).$$

Proof of Lemma 35 The result follows by taking $\epsilon = 1/2$ in Corollary A.1.14 of Alon and Spencer (2004).

Appendix E: Experimental Details

E.1. Synthetic Experiment Details

Offline. Each instance receives equally $n_j = 100$ observations (consider the standard setting, i.e., the arrival probability $p_j = 1/N$ for all $j \in [N]$). We generate the shared parameters β^\dagger by drawing each element independently from a uniform distribution $\text{Uniform}[0, 2]$, and normalizing them such that $\|\beta^\dagger\|_1 = 2$. We randomly draw s entries out of the d dimensions for each bias term δ^j independently, and then draw the values of the s nonzero entries independently from a uniform distribution $\text{Uniform}[0, 1]$. Next, we draw the context vectors X_t independently from a gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, truncated at -1 and 1 so that $\|X_t\|_\infty = 1$. We draw the noise ϵ_t independently from a gaussian distribution $\mathcal{N}(0, \sigma_j^2)$ with $\sigma_j = 0.05$ for any instance $j \in [N]$. To test the performance of our algorithm, we leave 20% of the data of the target instance as the test set, and use a 4-fold cross validation to tune all the hyperparameters on the rest 80%.

Online. Our total time horizon across instances T equals 15,000, 10,000 and 60,000 respectively for the three settings in Figure 6 and the arrival probability $p_j = 1/N$ for all $j \in [N]$; thus, each instance will receive an expected 500, 1,000 and 4,000 observations respectively. We generate the shared parameters $\{\beta_k^\dagger\}_{k \in [K]}$ by drawing each element independently from a uniform distribution $\text{Uniform}[0, 2]$, and normalizing them such that $\|\beta_k^\dagger\|_1 = 5$. We randomly draw s entries out of the d dimensions for each bias term δ_k^j independently, and then draw the values of the s nonzero entries independently from a uniform distribution $\text{Uniform}[0, 1]$. Next, we draw the context vectors X_t independently from a gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, truncated at -1 and 1 so that $\|X_t\|_\infty = 1$. We draw the noise ϵ_t independently from a gaussian distribution $\mathcal{N}(0, \sigma_j^2)$ with $\sigma_j = 0.05$ for all instances $j \in [N]$.

To ensure fair comparison, we tune the hyperparameters of all algorithms on a pre-specified grid. Define a hyperparameter q_0 to be such that $q = q_0 KN$ for q in our Algorithm 2. We take $q = 1$ for LASSO, OLS Bandit and the pooling algorithm, and $\lambda_1 = \lambda_{2,0} = 0.005$ for LASSO Bandit (note that the definition of q in Bastani and Bayati (2020) is different from ours). We take $\alpha = 0.5$ for GOBLin. We apply a trace-norm regularization on the parameters for each arm $k \in [K]$ respectively, and set the tuning constant in λ_n to be 0.005 in Trace-norm Bandit. For RMBandit, we take $q_0 = 0.5$; additionally, we set $\omega_0 = \eta_{1,0} = 0.2$, $\zeta_{1,0} = 0.1$, and $\lambda_0 = \lambda_{1,0} = 0.005$ for the first two settings (a) and (b), and $\omega_0 = \zeta_{1,0} = \eta_{1,0} = 0.05$ and $\lambda_0 = \lambda_{1,0} = 0.01$ for the third setting (c). Due to the setting difference, we take $h = 5$ for the first two settings and $h = 10$ for the third setting. For the robustness of our choices of the hyperparameters, please see additional experiments in Appendix F.

E.2. Diabetes Experiment Details

Our original dataset consists of 9,948 patients observed from 379 healthcare providers. However, many of these providers observe very few patients, so we restrict our experiment to the $N = 13$ largest hospitals, of which each has at least 150 unique patients (mean 317, median 301) observed during the sample period. We perform standard variable selection as a pre-processing step in order to avoid overfitting when computing our linear oracles. In particular, we run a LASSO variable selection procedure by regressing diabetes outcomes against the 184 total features (note that we exclude the healthcare providers that we use in our experiment in this step to avoid overfitting), and we tune the hyperparameters using 10-fold cross-validation. This leaves us with roughly 80 commonly predictive features (the number of selected features depends on the randomness in the cross-validation procedure). Note that this is still a relatively large number of features compared to the number of observations, supporting our argument that arm parameters are likely dense.

We fit a linear oracle to data from the target provider in hindsight; to avoid overfitting, we use a leave-one-out approach, i.e., for each patient, we train the best linear model on all data from the target provider excluding the current patient. Our oracle is constructed to provide the best achievable mean squared error within a linear model family. For the offline setting, we leave 50% of the data as the test set, and use a 4-fold cross validation to tune the hyperparameters on the rest data. For the online setting, to ensure fair comparison, we tune the hyperparameters of all algorithms, and we report the optimized results in Figure 7.

E.3. Pricing Experiment Details

Data. The original dataset covers 145 weeks of orders from 77 fulfillment centers across 51 cities. There are 14 different categories (e.g., beverages, snacks) and 4 different cuisines (e.g., Indian, Italian) for meals delivered by the company. We restrict our experiment to fulfillment centers in the three largest cities that have more than 2 fulfillment centers. Thus, we have $N = 20$ centers, each processing an average (median) of 6,747 (6,894) orders during the sample period. All 7 fulfillment centers are similarly sized (the smallest has 6072 orders and the largest has 7046 orders). One order arrives at each time step, and the chosen price is the checkout price, which includes discounts, taxes and delivery charges. The order price in our data ranges from \$55 to \$729; thus, we set $p_{\min} = 40$ and $p_{\max} = 800$. Following standard practice, we also normalize the price so it has a similar scale as the other features. Our outcome (demand) is given by the quantity in each order. The contexts are order-specific features including dummy variables capturing the category and cuisine, indicators of email or homepage promotions, and an intercept. Overall, our X_t has dimension 19, and therefore the dimensionality of the unknown parameters of the pricing model $d = 38$.

Algorithm. We now embed our robust multitask estimator within the ILSX/ILQX algorithmic framework proposed in Ban and Keskin (2021) to design our RMX algorithm; similarly, we embed the Laplacian estimator used by GOBLin (Cesa-Bianchi et al. 2013) to design the GOBX algorithm.

Let $\beta^j = \begin{bmatrix} \beta_0^j \\ \beta_1^j \end{bmatrix}$ denote the unknown parameters for instance j . For our forced samples, we fix two experimental prices $p_1 = 200$ and $p_2 = 600$, which we charge in two sets of periods

$$M_i^j = \left\{ t \mid \sum_{r \in [t]} \mathbb{1}(Z_r = j) = E^2 + i - 1, E = 1, 2, \dots \right\}$$

for each experimental price $i \in [2]$ and each instance $j \in [N]$. Note that M_i^j is a random set in the multitask setting, since it depends on the realization of arrivals Z_t . Let $M^j = M_1^j \cup M_2^j$ represent the forced price experimentation period, and let $M_t^j = \{r \mid r \in M^j, r < t\}$ be the set of time periods when prices are forced at instance j before time t . We update our RMEstimator of β^j 's at time periods

$$M = \{t \mid t = N(E^2 + 1), E = 1, 2, \dots\},$$

so that each instance obtains the same number of training observations in expectation as in the single-instance setting⁷. Then, the samples used for estimating the optimal price at time t are $\bigcup_{j \in [N]} M_{\gamma_t}^j$, where $\gamma_t = \max\{r \mid r \in M, r < t\}$.

Note that we now only maintain a single set of estimated parameters for instances (compared to both forced-sample and all-sample estimators in Algorithm 2). We denote our RMEstimator (Algorithm 1) at instance j at time t as

$$\hat{\beta}^j(\bigcup_{j \in [N]} M_{\gamma_t}^j, \lambda_{j,t}, \omega_t).$$

The first argument indicates the training data, i.e., all observations with price experimentation before time γ_t (recall that the robust multitask estimators are only updated at $t \in M$); the remaining arguments are hyperparameters. We denote the estimated optimal price of user X_t at instance j at time t as

$$\hat{p}^j(X_t, \hat{\beta}^j) = \frac{X_t^\top \hat{\beta}_0^j}{-2X_t^\top \hat{\beta}_1^j},$$

Algorithm 3 Robust Multitask Estimator with Price Experimentation (RMX)

Input parameters: Initial hyperparameters ζ_0, η_0, λ
Initialize $\{M_i^j\}_{i \in [2]}$, and M
for $t \in [T]$ **do**
 Observe an arrival at instance $j = Z_t$ and corresponding context X_t
 if $t \in M_i^j$ **then**
 Charge price $p_t = p_i$
 else
 Charge price $p_t = \hat{p}^j(X_t, \hat{\beta}^j(\cup_{j \in [N]} M_{\gamma_t}^j, \lambda_{j,t}, \omega_t))$
 end if
 Observe demand $Y_t = X_t^\top \beta_0^j + p_t \cdot (X_t^\top \beta_1^j) + \epsilon_t$
 if $t \in M$ **then**
 Update $\zeta_t = \zeta_0, \eta_t = \eta_0 \sqrt{\log(d \min_{j \in [N], |M_{\gamma_t}^j| > 0} |M_{\gamma_t}^j|)}$, and $\omega_t = \zeta_t + \eta_t$
 Update $\lambda_{j,t} = \lambda_{j,0} |M_{\gamma_t}^j|^{\frac{1}{4}} \sqrt{\log(d |M_{\gamma_t}^j|)}$ for each $j \in [N]$
 end if
end for

which is truncated at p_{\min} and p_{\max} . We formalize our algorithm in Algorithm 3.

The GOBX algorithm follows exactly as in Algorithm 3, but uses the Laplacian-regularized estimator from (Cesa-Bianchi et al. 2013) instead of our robust multitask estimator. Once again, to ensure fair comparison, we tune the hyperparameters of all algorithms, and we report the optimized results in Figure 8.

Appendix F: Additional Experiments

F.1. Dependence on Parameters in RMEstimator

First, we study how the performance of RMEstimator scales with the parameters in our model setup. In the following, we mainly focus on the setting (c) in Figure 4, where $N = 15$, $d = 40$, and $s = 5$. The model setup is the same as the offline setting described in §E.1.

Aligned with our theory, Figure 9 shows that the prediction error decreases with the number of instances N , increases with the sparsity level s and again decreases with the sample size ratio n_i/n_j for a target instance j ($n_i = n_{i'}$ for $i \neq i'$ and $i, i' \neq j$). Intuitively, when more auxiliary information is available, e.g., a larger number of instances or higher arrival rate in neighboring instances, our estimation or prediction becomes more accurate and hence the prediction error declines. However, when the instances become more heterogeneous, e.g., a higher sparsity level, less shared information can be transferred and the problem becomes harder to learn. We note that (c) in Figure 9 is related to the data-poor regime; in particular, the prediction error in a data-poor instance (i.e., $n_i/n_j \gg 1$) is lower than the corresponding prediction error in a standard instance (i.e., $n_i/n_j \approx 1$) for RMEstimator, consistent with Corollary 2.

In addition, we also study how RMEstimator reacts to different values of the parameters β^j 's. Figure 10 (a) shows the prediction error versus the alignment of δ^j 's. We randomly select s out of the first d_c dimensions, instead of out of all the d dimensions, to create δ^j 's. Thus, for smaller value of d_c , the supports of δ^j 's are more aligned. We find a non-monotonic relation between the prediction error and the value of d_c . Intuitively, RMEstimator performs better when there are fewer well-aligned components to learn together with δ^j in Step

⁷ We initialize our algorithm with OLS until each instance has at least observed 2 orders.

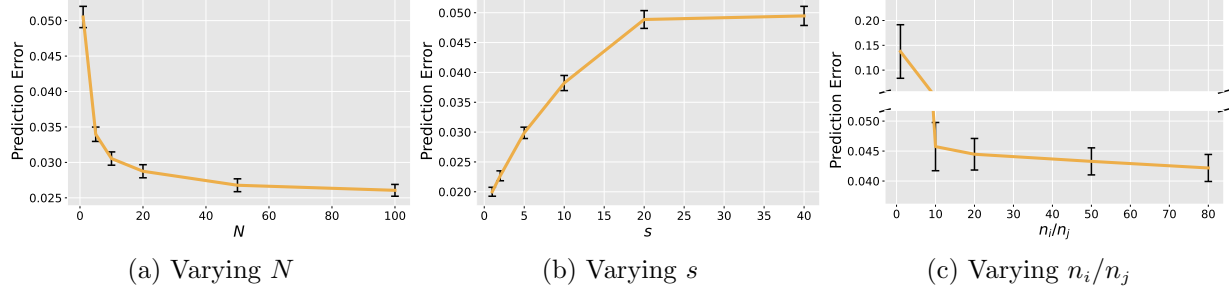


Figure 9 Lines depict the prediction error averaged over 200 trials of a single linear instance, with error bars the corresponding 95% confidence intervals.

2 (i.e., $|\mathcal{I}_{\text{well}}^\zeta|$ is smaller); that takes place either when the supports of δ^j 's are very aligned and $|\mathcal{I}_{\text{well}}^\zeta| \approx s$ or when the supports of δ^j 's are not aligned at all so each dimension is approximately poorly-aligned and $|\mathcal{I}_{\text{well}}^\zeta| \approx 0$. It is worth noting that even in the worst case, RMEstimator can still provide a reasonably good guarantee on the prediction error, which shows the robustness of this algorithm. Figure 10 (b) compares a pooling algorithm with RMEstimator given different magnitudes of δ^j 's. More specifically, we draw the values of the s nonzero entries of δ^j from a uniform distribution $\text{Uniform}[0, a]$. We find the pooling algorithm outperforms RMEstimator only when the value of a is very small. Nevertheless, even in that case, our algorithm still transfers most of the information and provides a competitive prediction accuracy as the pooling algorithm. This suggests our algorithm can be useful in most of the settings empirically. Finally, we analyze the performance of our algorithm when δ^j 's are approximately sparse in Figure 10 (c). In detail, we add a noise of $\text{Uniform}[-a_p, a_p]$ on each dimension of δ^j . Therefore, δ^j 's become less sparse when a_p takes larger values. The result is consistent with Figure 9 (b); that is, when a_p is larger and there is less shared structure across instances, the problem becomes harder and our prediction accuracy declines.

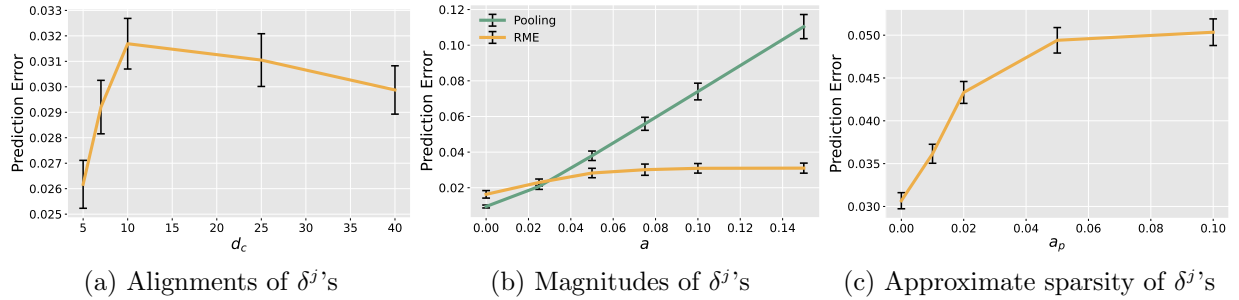


Figure 10 Lines depict the prediction error averaged over 200 trials of a single linear instance, with error bars the corresponding 95% confidence intervals.

F.2. Robustness to Hyperparameters in RMBandit

We now study the cumulative expected regret of RMBandit algorithm varying the hyperparameters specific to our algorithm: (i) $q_0 \in \{0.2, 0.5, 0.7, 1\}$ (the hyperparameter q_0 is such that $q = q_0 K N$ for q in Algorithm 2), (ii) $\omega_0, \zeta_{1,0}, \eta_{1,0} \in \{0.1, 0.2, 0.3\}$. In the following, we only focus on setting (b) in Figure 6, i.e., $N = 10$, $K = 10$, $d = 20$, and $s = 2$. The results in Figure 11 are calculated over $T = 10,000$ total time steps and

averaged over 20 trials. We find that the cumulative regret is not substantially affected considering varying values of these hyperparameters; this suggests that our algorithm is robust, which is important especially in empirical applications where these hyperparameters might not be well specified.

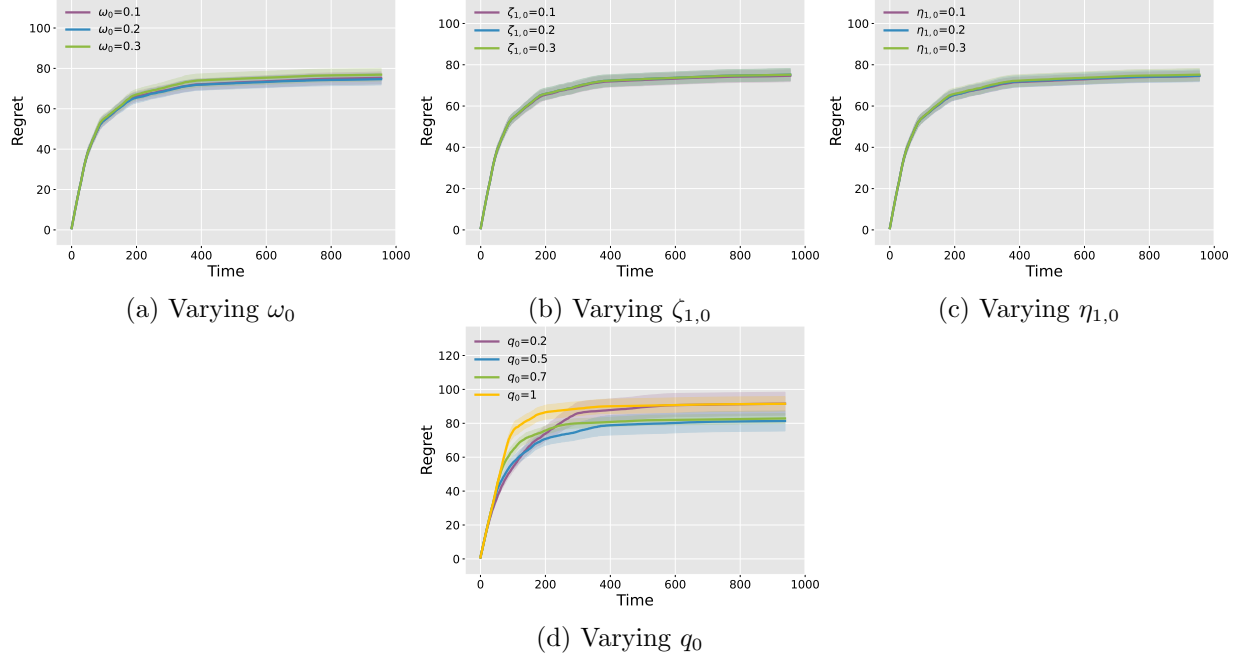


Figure 11 Lines depict the cumulative regret averaged over 20 trials of a single linear contextual bandit, with shaded regions the corresponding 95% confidence intervals.