

---

# Approximate Function Evaluation via Multi-Armed Bandits

---

Tavor Z. Baharav

Gary Cheng

Mert Pilanci

David Tse

Stanford University

## Abstract

We study the problem of estimating the value of a known smooth function  $f$  at an unknown point  $\boldsymbol{\mu} \in \mathbb{R}^n$ , where each component  $\mu_i$  can be sampled via a noisy oracle. Sampling more frequently components of  $\boldsymbol{\mu}$  corresponding to directions of the function with larger directional derivatives is more sample-efficient. However, as  $\boldsymbol{\mu}$  is unknown, the optimal sampling frequencies are also unknown. We design an instance-adaptive algorithm that learns to sample according to the importance of each coordinate, and with probability at least  $1-\delta$  returns an  $\varepsilon$  accurate estimate of  $f(\boldsymbol{\mu})$ . We generalize our algorithm to adapt to heteroskedastic noise, and prove asymptotic optimality when  $f$  is linear. We corroborate our theoretical results with numerical experiments, showing the dramatic gains afforded by adaptivity.

## 1 Introduction

Estimation is a ubiquitous and expensive task in many modern applications. Often, however, we are not interested in estimation for estimation’s sake, but rather for a downstream task through an application specific function. For an unknown parameter vector  $\boldsymbol{\mu} \in \mathbb{R}^n$  which we are given noisy coordinate-wise query access to, it is well understood how to sample to construct an estimate to minimize the approximation error between our estimate  $\hat{\boldsymbol{\mu}}$  and the true vector  $\boldsymbol{\mu}$ . Instead, if we are interested in estimating  $f(\boldsymbol{\mu})$ , a natural question is whether we can leverage knowledge of this function  $f$  to construct a more efficient estimation procedure. In this paper we show that using adaptivity we can

dramatically reduce the number of samples required to estimate  $f(\boldsymbol{\mu})$  to a target accuracy  $\varepsilon$ , exploiting the fact that we are interested in  $\boldsymbol{\mu}$  solely through its evaluation  $f(\boldsymbol{\mu})$ . This problem has been well studied in the special case of  $f(\boldsymbol{\mu}) = \max_{i \in [n]} \mu_i$  in the multi-armed bandit literature. In this paper, we focus on settings where  $f$  has Lipschitz gradients.

We motivate our focus on smooth  $f$  with several applications; software reliability testing [Cai et al., 2004], stratified sampling for Monte Carlo integration [Carpentier et al., 2015], and estimating the norm of a matrix vector product  $\|Ax\|_2^2$ . In software reliability testing [Cai et al., 2004], we are interested in estimating the failure rate of a piece of software across  $n$  classes or tasks. This is formulated as estimating  $f(\boldsymbol{\mu}) = \sum_{i=1}^n p_i \mu_i$ , where  $\mu_i$  is the probability that a test from class  $i$  fails, and  $p_i$  is the fraction of time that a task in class  $i$  will be used. Since  $\boldsymbol{\mu}$  is unknown a priori, we adaptively select what subroutines to test in order to estimate  $f(\boldsymbol{\mu})$ . We note that in this setting the difficulty of the problem stems not from the complexity of  $f$ , as it is linear, but rather from the unknown variance of our observations. These variances amplify the estimation errors in different coordinates by differing amounts, and so the optimal sampling frequencies critically depend on these unknown variances

Another application of smooth function estimation lies in the basic linear algebraic task of estimating the power in a matrix vector product  $\|Ax\|_2^2$ . For  $A \in \mathbb{R}^{n \times d}$ , we see that defining  $\boldsymbol{\mu} \triangleq Ax$  we have that  $dA_{i \cdot} J x_J$  is an unbiased estimator of  $A_i^\top x$  when  $J \sim \text{Unif}([d])$ . This primitive of converting a computational problem to one of statistical estimation was formalized in [Bagaria et al., 2021]. This allows us to convert the computational problem of estimating  $\|Ax\|_2^2$  into the problem of estimating  $f(\boldsymbol{\mu}) = \|\boldsymbol{\mu}\|_2^2$  for an unknown vector  $\boldsymbol{\mu}$  that can only be sampled coordinate-wise via a noisy oracle.

We operate in the stochastic multi-armed bandit setting, a common framework for studying sequential decision making problems [Lai and Robbins, 1985]. In this setting the vector of arm means  $\boldsymbol{\mu} \in$

---

Proceedings of the 25<sup>th</sup> International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

$\mathbb{R}^n$  is fixed but unknown, and at every time step we are able to query one coordinate of  $\mu_i$ , an arm, and obtain a noisy but unbiased sample, an arm pull [Bubeck and Cesa-Bianchi, 2012]. A prototypical multi-armed bandit problem is best arm identification, which has been extensively studied [Jamieson and Nowak, 2014], and can be posed as  $f(\boldsymbol{\mu}) = \operatorname{argmax}_{i \in [n]} \mu_i$  where  $f: \mathbb{R}^n \mapsto [n]$  for  $[n] \triangleq \{1, \dots, n\}$ . With this discrete valued output, the error between  $f(\boldsymbol{\mu})$  and  $f(\hat{\boldsymbol{\mu}})$  is taken as 0 if they are equal and 1 otherwise, with the same 0-1 loss being used across other pure exploration multi-armed bandit variants [Lattimore and Szepesvári, 2020].

For smooth  $f$ , however, we are better able to analyze the function error caused by our  $\boldsymbol{\mu}$  estimation error. We obtain via a Taylor expansion that a plug-in estimator  $\hat{\boldsymbol{\mu}}$  will achieve an error of  $f(\hat{\boldsymbol{\mu}}) - f(\boldsymbol{\mu}) = \nabla f(\boldsymbol{\mu})^\top (\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}) + O(\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_2^2)$ . This suggests that in the high accuracy regime where  $\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_2$  is small, it is beneficial to sample coordinates with larger partial derivatives more. Such Taylor expansions do not arise in classical multi-armed bandit variants like the best arm identification problem, as examining its real valued analogue  $f(\boldsymbol{\mu}) = \max_{i \in [n]} \mu_i$ , we see that  $\nabla f(\boldsymbol{\mu}) = e_{i^*}$ , where  $i^* = \operatorname{argmax}_{i \in [n]} \mu_i$  (assumed to be unique). If  $\boldsymbol{\mu}$  is well estimated such that  $i^*$  has been identified, then the only way to improve our estimation of  $f(\boldsymbol{\mu})$  is to improve our estimation of  $\mu_{i^*}$ . This agrees with intuition but is not very useful, as in order for this dependence to hold, we need to have identified  $i^*$ , as  $f$  is not everywhere differentiable nor are its gradients smooth. In the setting with smooth  $f$  however, we can guarantee that our estimate of the gradient  $\nabla f(\boldsymbol{\mu})$  will improve as our estimate of  $\boldsymbol{\mu}$  improves.

### 1.1 Outline

In Section 2 we discuss related work and place this paper within the broader multi-armed bandit literature. We formalize our problem setting in Section 3. In Section 4 we propose our algorithm for adaptive function approximation. We extend our analysis to more general scenarios in Section 5. In Section 6 we numerically validate the theoretical improvements afforded by our scheme. We conclude in Section 7. All theoretical claims are proved in the Appendix.

## 2 Related work

The study of multi-armed bandits has a long history dating back to the 1950’s [Robbins, 1952, Paulson, 1964, Lai and Robbins, 1985]. Numerous variants of this fundamental problem have been studied, including many identification-based objectives

[Lattimore and Szepesvári, 2020]. Regarding estimation of a known functional of the unknown mean vector, several previous works have studied the case of linear functions  $f$  with heteroskedastic noise, which we highlight below. While the aim of our work is on constructing an algorithm with a stopping condition and fixed confidence guarantees, these alternative schemes instead focus on regret and anytime performance and do not provide such guarantees.

### 2.1 Software reliability assessment

One well studied problem that fits as a special case of our results is software reliability testing [Cai et al., 2004]. In this setting there are  $n$  classes of tests  $C_1, \dots, C_n$  that can be run on a piece of software. Tests from class  $i$  fail independently with an unknown but fixed probability  $\mu_i$ . The system is known to be run under a given operational profile  $\{p_i\}_{i=1}^n$ , where  $p_i$  indicates the fraction of time that a task in class  $C_i$  will be used. In this setting the objective is to estimate the unreliability of the system, which is  $f(\boldsymbol{\mu}) = \mathbf{p}^\top \boldsymbol{\mu}$ . Works in the adaptive software reliability estimation literature utilize the plug-in estimator  $f(\hat{\boldsymbol{\mu}})$  which is unbiased due to the linearity of  $f$ , and so the primary objective is to minimize the estimator’s variance. This is accomplished by a careful finite sample analysis of the corresponding Markov Chain that is induced by all possible sampling patterns [Hu et al., 2013]. Since each arm fails with probability  $\mu_i$  and succeeds with probability  $1 - \mu_i$ , the noise in this problem is heteroskedastic. Asymptotic sampling frequencies are derived by using a Lagrangian for constrained optimization [Lv et al., 2014]. These works provide a specialized analysis for the setting with Bernoulli noise, and can only accommodate linear functions  $f$ .

### 2.2 Stratified sampling bandit literature

A similar vein of literature has arisen in works on adaptive stratified sampling for Monte-Carlo integration [Carpentier and Munos, 2012]. In this setting the objective is to integrate a function  $f$  over a domain  $\mathcal{X}$  by partitioning the domain into strata and adaptively allocating samples within. Assuming that we can compute the measure of each strata  $i = 1, \dots, n$ , the objective can be formulated as estimating  $f(\boldsymbol{\mu}) = \sum_i w_i \mu_i$ , where  $w_i$  is the measure of strata  $i$  and  $\mu_i$  is the expectation of  $f$  over strata  $i$ . Due to the different and unknown variances  $\sigma_i^2$  of  $f$  over different strata, adaptivity is required to learn the optimal sampling frequencies  $\alpha_i \propto w_i \sigma_i$ , the well known Neyman allocation [Lohr, 2019], in order to obtain sublinear regret. An important distinction between the Monte Carlo integration problem and our setting is that in these works the objective is to estimate  $\mathbb{E}\{f(X)\}$ . In our

setting however, we are interested in evaluating  $f$  at some ground truth value  $\boldsymbol{\mu} = \mathbb{E}\{X\}$  of which we are only able to obtain noisy coordinate-wise observations. This means that our objective is to estimate  $f(\mathbb{E}\{X\})$ , which only coincides with the previous objective when  $f$  is linear.

As these stratified sampling works both focus on regret, they provide guarantees on the deviation of the empirical sampling ratios from the optimal ones. However, it is unclear how to construct algorithmic stopping conditions for these regret-focused methods to allow them to operate in the fixed confidence regime. Such stopping conditions are necessary for fixed confidence algorithms, which are significantly more useful in practical estimation settings. In the software reliability estimation setting for example, we wish to run tests until we are 98% sure that we’ve estimated the reliability to within 1%; regret is not a notion of interest, as the primary goal is this fixed confidence guarantee. Additionally these works can only handle linear  $f$  with heteroskedastic noise. We are able to generalize to arbitrary differentiable  $f$  with  $L$ -Lipschitz gradients, providing fixed confidence guarantees.

### 2.3 Bandits as a computational tool

Due to the ever increasing size of datasets, multi-armed bandit-based randomized algorithms have been recognized as a useful tool for constructing instance-optimal algorithms for computational tasks. Dating back to works in Monte Carlo Tree Search [Kocsis and Szepesvári, 2006] and hyperparameter tuning [Li et al., 2017], adaptivity has been used to focus computational efforts towards promising solutions. Formalized into the framework of Bandit-Based Monte Carlo Optimization [Bagaria et al., 2021], this reduction from computation to adaptive statistical estimation has been used to solve many problems including finding the medoid of a dataset [Bagaria et al., 2018, Baharav and Tse, 2019, Tiwari et al., 2020],  $k$ -nearest neighbor graph construction [LeJeune et al., 2019, Mason et al., 2019, Mason et al., 2021], Monte Carlo permutation-based multiple testing [Zhang et al., 2019], and mode estimation [Singhal et al., 2021]. Most similarly, a rank-one estimation problem was solved in [Kamath et al., 2020] via an adaptive SVD, showing the utility of this technique for linear algebraic primitives, which we extend here to estimating  $\|Ax\|_2^2$ .

In order to realize statistical gains as wall-clock gains, it is advantageous to use algorithms that require fewer rounds of adaptivity and avoid extra synchronization steps. Semi-adaptive algorithms have been studied in various contexts in the multi-armed bandit literature, first for a multiplayer setting

[Hillel et al., 2013], then for halving based algorithms [Karnin et al., 2013], and recently in more general settings [Karpov and Zhang, 2020, Karbasi et al., 2021], including for submodular optimization [Balkanski and Singer, 2018, Esfandiari et al., 2021]. Such schemes exploit the efficiency of batched operations in modern computer architectures leading to dramatically faster algorithms [Baharav and Tse, 2019, Bagaria et al., 2021]. To this end, all algorithms proposed in this paper perform batched sampling, and show dramatic wall-clock improvements over fully sequential methods in our experiments.

## 3 Problem Formulation

We study the problem of minimizing the number of samples  $T$  required to construct an estimator  $\hat{\boldsymbol{\mu}}$  of  $\boldsymbol{\mu}$  such that  $|f(\boldsymbol{\mu}) - f(\hat{\boldsymbol{\mu}})| \leq \varepsilon$  with probability at least  $1 - \delta$ . The function  $f : \mathbb{R}^n \mapsto \mathbb{R}$  is known, and is differentiable with  $L$ -Lipschitz gradients, where we denote  $\nabla_i f(\mathbf{x}) \triangleq \frac{\partial}{\partial x_i} f(\mathbf{x}) \triangleq g_i$ .  $\boldsymbol{\mu}$  is unknown, and at each time step  $t \in \mathbb{N}$  we select  $i \in [n]$  and obtain  $\mu_i + Z_{i,t}$ , where  $\{Z_{i,t}\}$  are independent standard normals. Our results are readily extendable to the case where  $\{Z_{i,t}\}$  are 1-sub-Gaussian, but for clarity of exposition we present our results for the Gaussian case. Given these three metrics of interest  $(\varepsilon, \delta, T)$ , we focus on algorithms for the fixed confidence setting where we constrain both the error probability  $\delta$  and the output error  $\varepsilon$ , and minimize the number of samples needed.

## 4 Adaptive Sampling Scheme

To design our adaptive algorithm, we begin by examining the error incurred by a plug-in estimator  $\hat{\boldsymbol{\mu}}$  more closely. Due to the  $L$ -Lipschitz gradients of  $f$  [Bubeck, 2013] this error satisfies

$$|f(\hat{\boldsymbol{\mu}}) - f(\boldsymbol{\mu}) - \nabla f(\boldsymbol{\mu})^\top (\hat{\boldsymbol{\mu}} - \boldsymbol{\mu})| \leq \frac{L}{2} \|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_2^2. \quad (1)$$

Since our estimator for  $\mu_i$  is distributed as  $\hat{\mu}_i \sim \mathcal{N}(\mu_i, 1/T_i)$  after  $T_i$  samples, the second order error  $\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_2^2$  is readily bounded by uniformly sampling the arms. This indicates that we need to have a sufficiently accurate estimate of  $\boldsymbol{\mu}$  before our linearized first order approximation is accurate. Once the second order error has been sufficiently controlled, we examine the first order error  $\nabla f(\boldsymbol{\mu})^\top (\hat{\boldsymbol{\mu}} - \boldsymbol{\mu})$  by noting that

$$\nabla f(\boldsymbol{\mu})^\top (\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}) \sim \mathcal{N}\left(0, \sum_{i=1}^n \frac{g_i^2}{T_i}\right). \quad (2)$$

To guarantee that this error is less than  $\varepsilon$  in magnitude with probability at least  $1 - \delta$ , Hoeffding’s inequality

indicates that we need to sample until  $\sum_{i=1}^n g_i^2/T_i \leq 2\varepsilon^2/\log(2/\delta)$ . Minimizing our sample complexity in order to reach this condition, we define our sampling distribution  $\alpha \in \Delta^n$ , the  $n$  dimensional probability simplex, such that  $T_i = \alpha_i T$ . Observe that for a fixed number of samples  $T$  we can equivalently minimize over  $\alpha$  instead of  $\{T_i\}$ , yielding the optimal sampling distribution

$$\operatorname{argmin}_{\alpha \in \Delta^n} \sum_i \frac{g_i^2}{\alpha_i T} \implies \alpha_i \propto |g_i|, \quad (3)$$

due to Sion's minimax theorem [Sion, 1958] and first order conditions. Sampling each coordinate proportional to the magnitude of its partial derivative  $|g_i|$  effectively rescales our confidence interval on each arm inversely proportional to the scalar multiplying it.

If  $\mathbf{g}$  is known a priori, as is the case in the linear setting, then so is the optimal allocation, and hence adaptivity is not necessary. In our setting however, the gradient  $\mathbf{g} = \nabla f(\boldsymbol{\mu})$  is unknown, as the point  $\boldsymbol{\mu}$  is unknown. Motivated by the goal of sampling proportionally to the unknown gradient magnitudes, we propose Algorithm 1. It proceeds in rounds with doubling budgets  $B_r = 2^r B_0$ , and within each round uniformly samples the arms to yield improved estimation of  $\boldsymbol{\mu}$  and by extension  $\nabla f(\boldsymbol{\mu})$ , to estimate the optimal sampling frequencies. At each iteration the algorithm checks the stopping condition in Line 10, determining whether sampling according to the estimated optimal sampling frequencies with the same budget would yield sufficiently small error. If so, it performs these pulls and returns the plug-in estimate of  $f(\boldsymbol{\mu})$ . Otherwise, it continues to the next round to refine its estimates of the optimal sampling frequencies.

Through our algorithm we construct estimators  $\hat{\boldsymbol{\mu}}^{(r)}$  of  $\boldsymbol{\mu}$  for estimating  $\nabla f(\boldsymbol{\mu})$ , where  $\hat{\mu}_i^{(r)}$  is the average of the  $\tilde{B}_r \triangleq \sum_{\ell=1}^r \lceil B_\ell/n \rceil$  samples of arm  $i$  taken prior to and including round  $r$ . We utilize anytime confidence intervals of width  $C_r$ , where after  $\tilde{B}_r$  pulls of arm  $i$  we have that

$$|\hat{\mu}_i^{(r)} - \mu_i| \leq \sqrt{\frac{2 \log(12nr^2/\delta)}{\tilde{B}_r}} \triangleq C_r \quad (4)$$

with high probability, formalized in Lemma 1. While tighter confidence intervals based on the law of the iterated logarithm can be utilized as in [Jamieson et al., 2014, Kaufmann et al., 2016], in this work we use simple ones derived from Hoeffding's inequality for clarity of exposition.

To estimate the importance of each coordinate, we construct upper and lower bounds for our gradient as

$$\begin{aligned} \hat{g}_i^{(r,L)} &\triangleq \min_{\mathbf{y}: \|\mathbf{y} - \hat{\boldsymbol{\mu}}^{(r)}\|_\infty \leq C_r} |\nabla_i f(\mathbf{y})|, \\ \hat{g}_i^{(r,U)} &\triangleq \max_{\mathbf{y}: \|\mathbf{y} - \hat{\boldsymbol{\mu}}^{(r)}\|_\infty \leq C_r} |\nabla_i f(\mathbf{y})|. \end{aligned} \quad (5)$$

Assuming that our confidence intervals hold, we have that  $\hat{g}_i^{(r,L)} \leq g_i \leq \hat{g}_i^{(r,U)}$  for all  $i, r$ . Our estimators  $\hat{g}_i^{(r,L)}$  presume that the partial derivative is as small as it could feasibly be. This is critical to ensure that we do not oversample unimportant arms. If this minimization is computationally intensive, one efficient alternative is to define  $\hat{g}_i^{(r,L)}$  as  $(|\nabla_i f(\hat{\boldsymbol{\mu}}^{(r)})| - L\sqrt{n}C_r)_+$  where  $(x)_+ \triangleq \max(x, 0)$ . This coarse bound ignores the local geometry of the problem and instead utilizes the global Lipschitzness bound  $L$ , but maintains the same guarantees for the worst case functions. We correspondingly can set  $\hat{g}_i^{(r,U)}$  as  $|\nabla_i f(\hat{\boldsymbol{\mu}}^{(r)})| + L\sqrt{n}C_r$ . These overestimates  $\hat{g}_i^{(r,U)}$  ensure that we do not terminate too early.

---

**Algorithm 1** Adaptive function approximation
 

---

- 1: **Input:** arms  $[n]$ , function  $f$ , target accuracy  $\varepsilon$ , error probability  $\delta$
  - 2: Set  $B_0 = 17Ln^2\varepsilon^{-1}\log(12n/\delta)$
  - 3: **for**  $r = 1, 2, \dots$  **do**
  - 4:     Set round budget  $B_r = B_0 2^r$
  - 5:     Pull each arm  $\lceil B_r/n \rceil$  times
  - 6:     Use all previous samples to construct  $\hat{\boldsymbol{\mu}}^{(r)}$
  - 7:     Compute bounds  $\{\hat{g}_i^{(r,L)}, \hat{g}_i^{(r,U)}\}$  as in (5)
  - 8:     Construct sampling frequencies  $\alpha_i^{(r)} \propto \hat{g}_i^{(r,L)}$
  - 9:     Set  $\tilde{T}_i^{(r)} \triangleq \lceil (\alpha_i + \frac{1}{n})B_r \rceil$
  - 10:    **if**  $\sum_i \frac{(\hat{g}_i^{(r,U)})^2}{\tilde{T}_i^{(r)}} \leq \frac{\varepsilon^2}{16 \log(6/\delta)}$  **then**
  - 11:       Pull arm  $i$   $\tilde{T}_i^{(r)}$  times to construct  $\tilde{\boldsymbol{\mu}}$
  - 12:       **return**  $f(\tilde{\boldsymbol{\mu}})$
  - 13:    **end if**
  - 14: **end for**
- 

We provide the following theorem regarding the performance of Algorithm 1. We express our results utilizing big O notation to highlight the dependence on relevant problem parameters. All constants are made explicit in Appendix B.

**Theorem 1.** *Algorithm 1 succeeds with probability at least  $1 - \delta$  in outputting  $\tilde{\boldsymbol{\mu}}$  such that  $|f(\tilde{\boldsymbol{\mu}}) - f(\boldsymbol{\mu})| \leq \varepsilon$ , using a number of arm pulls at most*

$$O\left(\frac{\|\nabla f(\boldsymbol{\mu})\|_1^2 \log(1/\delta)}{\varepsilon^2} + \frac{n^2 L \log(n/\delta)}{\varepsilon}\right).$$

The proof of this theorem is relegated to Appendix B, but we provide a sketch below. The proof begins by showing that the good event  $\xi$ , where the mean estimators  $\{\hat{\mu}_i^{(r)}\}$  stay within their confidence intervals, occurs with high probability. Then, we show that our error due to the second order terms is less than  $\varepsilon/2$  due to our choice of a sufficiently large  $B_0$ . Next, we show that our sampling frequencies  $\alpha_i^{(r)}$  are sufficiently

large for important arms. Critically, arms with small  $g_i$  are sufficiently sampled from the uniform sampling in  $\tilde{T}_i^{(r)}$ , and arms with large  $g_i$  will have large  $\alpha_i^{(r)}$ . Finally, we bound the number of rounds required until the termination condition is met.

In Appendix B.3 we discuss how Algorithm 1 and its analysis can be improved for the special case where  $f(\boldsymbol{\mu}) = \|\boldsymbol{\mu}\|_2^2$ , where we are able to tighten our bounds on the lower order terms. In this case we are able to improve our gradient bounds, defining  $\hat{g}_i^{(r,L)} \triangleq (|\hat{\mu}_i^{(r)}| - 4C_r)_+$ , where we no longer need to use the Lipschitzness of the entire gradient operator due to the separability of  $f$ . More importantly, we are now able to subtract the mean of the second order error terms; concretely  $\|\tilde{\boldsymbol{\mu}}^{(r)}\|_2^2 - \sum_{i=1}^n 1/\tilde{T}_i$  is now an unbiased estimator of  $f(\boldsymbol{\mu})$ . This allows us to argue probabilistically that our second order errors average out, reducing our lower order sample complexity by a factor of  $\sqrt{n}$ .

**Corollary 1.** *This variant of Algorithm 1 will with probability at least  $1 - \delta$  estimate  $\|\boldsymbol{\mu}\|_2^2$  to accuracy  $\varepsilon$  using a number of arm pulls at most*

$$O\left(\frac{\|\nabla f(\boldsymbol{\mu})\|_1^2 \log(1/\delta)}{\varepsilon^2} + \frac{n^{3/2} \log(n/\delta)}{\varepsilon}\right).$$

#### 4.1 Gain of adaptivity

Examining the result of Theorem 1, we see that the first term stemming from the first order Taylor expansion will dominate when  $H^2 > Ln^2\varepsilon$ , where the problem complexity measure  $H$  is defined as  $H \triangleq \|\nabla f(\boldsymbol{\mu})\|_1$ . This scaling with  $n$  is to be expected, as  $H$  is the sum of  $n$  terms. This naturally occurs in the high accuracy regime when  $\varepsilon \rightarrow 0$ , where we show that a sample complexity of  $\Theta(H^2 \log(1/\delta)\varepsilon^{-2})$  is the optimal dependence on problem parameters, as formalized in our lower bound in Section 5.3 (Proposition 1).

Comparing against the uniform sampling baseline, we see that taking  $T_i = T/n$  would require a budget of approximately  $n\|\nabla f(\boldsymbol{\mu})\|_2^2 \log(2/\delta)\varepsilon^{-2}$  pulls. Hence, the ratio between the number of samples required by this uniform baseline and the number of samples required by our adaptive scheme scales as

$$\text{gain}(f; \boldsymbol{\mu}) \triangleq \frac{n\|\nabla f(\boldsymbol{\mu})\|_2^2}{\|\nabla f(\boldsymbol{\mu})\|_1^2}, \quad (6)$$

which is bounded between 1 and  $n$ . This lower limit is achieved if all the partial derivatives are equal, as then there is no imbalance in the gradient to exploit. On the other extreme, if our gradient has only one nonzero entry we see that uniform sampling requires a factor of  $n$  more samples than its adaptive counterpart. Studying the special case of  $\ell_p$  norms, we see that

$\text{gain}(\|\cdot\|_1; \boldsymbol{\mu}) = 1$  as all coordinates matter equally, and  $\text{gain}(\|\cdot\|_2^2; \boldsymbol{\mu}) = n\|\boldsymbol{\mu}\|_2^2/\|\boldsymbol{\mu}\|_1^2 \geq 1$ . In accordance with intuition, in Appendix A we show that when the coordinates of  $\boldsymbol{\mu}$  are not all identical the gain of adaptivity in estimating  $\|\boldsymbol{\mu}\|_p^p$  is a monotonically increasing function of  $p \in [1, \infty)$ .

## 5 Algorithmic Extensions

We now discuss practical extensions to Algorithm 1, showing how it can be generalized to accommodate broader scenarios of practical interest.

### 5.1 Thresholding variant

We first extend our algorithm to the setting where our objective is to minimize the number of samples required to determine whether our function value  $f(\boldsymbol{\mu})$  is above a given threshold  $\tau$  or not, i.e. to determine whether  $f(\boldsymbol{\mu}) > \tau$ . This is a useful primitive in settings where the desired task is not estimation but rather decision making, for example in certifying software reliability. This thresholding objective can be thought of as providing an “adaptive”  $\varepsilon$ , where fewer samples can be used if  $f(\boldsymbol{\mu})$  is far from  $\tau$ . Our proposed algorithm for this setting proceeds in a similar manner to Algorithm 1, but at each round now explicitly computes an estimate  $f(\tilde{\boldsymbol{\mu}}^{(r)})$  to check whether the estimated gap  $|f(\tilde{\boldsymbol{\mu}}^{(r)}) - \tau|$  is large enough relative to a bound on the error  $|f(\tilde{\boldsymbol{\mu}}^{(r)}) - f(\boldsymbol{\mu})|$ .

---

#### Algorithm 2 Adaptive thresholding

---

- 1: **Input:** arms  $[n]$ , function  $f$ , threshold  $\tau$ , error probability  $\delta$
  - 2: **for**  $r = 1, 2, \dots$  **do**
  - 3:     Set round budget  $B_r = n2^r$
  - 4:     Pull each arm  $\lceil B_r/n \rceil$  times
  - 5:     Use all previous samples to construct  $\hat{\boldsymbol{\mu}}^{(r)}$
  - 6:     Compute bounds  $\{\hat{g}_i^{(r,L)}, \hat{g}_i^{(r,U)}\}$  as in (5)
  - 7:     Construct sampling frequencies  $\alpha_i^{(r)} \propto \hat{g}_i^{(r,L)}$
  - 8:     Set  $\tilde{T}_i^{(r)} \triangleq \lceil (\alpha_i + \frac{1}{n})B_r \rceil$
  - 9:     Pull arm  $i$   $\tilde{T}_i^{(r)}$  times to construct  $\tilde{\boldsymbol{\mu}}^{(r)}$
  - 10:    **if**  $C_r^{(f)} \leq |f(\tilde{\boldsymbol{\mu}}^{(r)}) - \tau|$  **then**
  - 11:        Define estimate  $\tilde{\boldsymbol{\mu}} \triangleq \tilde{\boldsymbol{\mu}}^{(r)}$
  - 12:        **return**  $f(\tilde{\boldsymbol{\mu}}) > \tau$
  - 13:    **end if**
  - 14: **end for**
- 

For this adaptive thresholding algorithm we define and utilize confidence intervals  $C_r^f$  for the function evaluation estimators  $f(\tilde{\boldsymbol{\mu}}^{(r)})$  such that  $|f(\tilde{\boldsymbol{\mu}}^{(r)}) - f(\boldsymbol{\mu})| \leq C_r^f$

with high probability, where

$$\begin{aligned} C_{r,1}^f &\triangleq \sqrt{2 \log(24nr^2/\delta) \sum_i \frac{(\hat{g}_i^{(r,U)})^2}{\tilde{T}_i^{(r)}}} \\ C_{r,2}^f &\triangleq \frac{Ln \log(24nr^2/\delta)}{\tilde{B}_r} \\ C_r^f &\triangleq C_{r,1} + C_{r,2}, \end{aligned} \quad (7)$$

where  $C_{r,1}^f$  is our confidence interval on our first order error in round  $r$ , and  $C_{r,2}^f$  is our confidence interval on our higher order error in round  $r$ . For simplicity we assume that  $f(\boldsymbol{\mu}) \neq \tau$ , but Algorithm 2 can be modified to output  $\perp$  in borderline cases where  $f(\boldsymbol{\mu})$  is very close to  $\tau$ . Concretely, the algorithm can output  $\perp$  if it has not yet terminated and  $C_r^f \leq \varepsilon/2$  for some input tolerance  $\varepsilon$  to indicate that  $|f(\boldsymbol{\mu}) - \tau| \leq \varepsilon$  with high probability. This is similar to the indifference zone formulation in the operations research community [Kim and Nelson, 2001]. We provide the following theorem regarding the performance of Algorithm 2, where  $\tilde{O}$  suppresses  $\log \log$  terms in  $\text{poly}(n, \delta, \|\nabla f(\boldsymbol{\mu})\|_1, (f(\boldsymbol{\mu}) - \tau)^{-1})$ .

**Theorem 2.** *Algorithm 2 succeeds with probability at least  $1 - \delta$  and outputs whether  $f(\boldsymbol{\mu}) > \tau$ , using*

$$\tilde{O} \left( \frac{\|\nabla f(\boldsymbol{\mu})\|_1^2 \log(1/\delta)}{(f(\boldsymbol{\mu}) - \tau)^2} + \frac{n^2 L \log(n/\delta)}{|f(\boldsymbol{\mu}) - \tau|} \right)$$

arm pulls, assuming that  $f(\boldsymbol{\mu}) \neq \tau$ .

The proof of this theorem, relegated to Appendix C, proceeds similarly to that of Theorem 1, with some interesting nuances stemming from the fact that our  $\varepsilon$  is now unknown. Since we do not a priori know what accuracy to estimate  $f(\boldsymbol{\mu})$  to, we need to sample according to our estimated optimal frequencies  $\alpha_i^{(r)}$  in every round to generate  $f(\tilde{\boldsymbol{\mu}}^{(r)})$  to estimate the necessary accuracy as  $|f(\tilde{\boldsymbol{\mu}}^{(r)}) - \tau|$ . Proceeding in rounds of doubling budgets, we analyze our sample complexity by bounding the number of rounds until our sampling frequencies are sufficiently accurate, our estimate of  $\varepsilon$  is sufficiently tight, and the number of samples is sufficiently large to yield the desired error.

## 5.2 Adapting to unknown variances

As stated, Algorithms 1 and 2 assume that the additive noise corrupting the arm pulls is i.i.d.  $\mathcal{N}(0, 1)$ . In many scenarios of practical interest however, noise variance may vary dramatically across arms. In software reliability assessment for example, the arm reward distributions are Bernoulli with mean  $\mu_i$ , where the variances are a function of the unknown arm rewards, with  $\sigma_i^2 = \mu_i(1 - \mu_i)$ . More generally, these un-

known variances can be thought of as nuisance parameters, which we need to estimate in order to efficiently run our algorithm but do not need to output. To adapt to this scenario we generalize our algorithm to accommodate bounded distributions by utilizing an empirical Bernstein-type bound [Maurer and Pontil, 2009], subsuming the software reliability assessment setting. Our doubly adaptive algorithm adapts to both the unknown arm variances  $\sigma_i$  as well as the unknown gradient  $\nabla f(\boldsymbol{\mu})$ . For simplicity of exposition, in this section we assume that our function is separable, i.e.  $f(\boldsymbol{\mu}) = \sum_{i=1}^n f_i(\mu_i)$ , but all ideas extend to the general case as discussed at the end of this section.

In order to accommodate these unknown variances, our new algorithm utilizes variance adaptive confidence intervals of width  $C_\mu(i, r)$  for our sample mean estimators  $\hat{\mu}_i^{(r)}$  of  $\mu_i$  in round  $r$ , defined in (9). To quantify the improved rate of decay of our confidence intervals, and to sample coordinates with high variance more, we utilize confidence intervals for our sample standard deviation estimators  $\hat{\sigma}_i^{(r)}$ . We construct these confidence intervals  $C_\sigma(i, r)$  for our estimator  $\hat{\sigma}_i^{(r)}$  of  $\sigma_i$  in round  $r$  as below:

$$C_\sigma(i, r) \triangleq \sqrt{\frac{2 \log(8nr^2/\delta)}{T_i^{(r)} - 1}} \quad (8)$$

$$C_\mu(i, r) \triangleq \hat{\sigma}_i^{(r)} \sqrt{\frac{2 \log(32nr^2/\delta)}{T_i^{(r)}} + \frac{7 \log(32nr^2/\delta)}{3(T_i^{(r)} - 1)}}. \quad (9)$$

In accordance with our variance adaptive confidence intervals, we redefine our gradient estimators as

$$\begin{aligned} \hat{g}_i^{(r,L)} &\triangleq \min_{|y_i - \hat{\mu}_i^{(r)}| \leq C_\mu(i, r)} |f'_i(y_i)|, \\ \hat{g}_i^{(r,U)} &\triangleq \max_{|y_i - \hat{\mu}_i^{(r)}| \leq C_\mu(i, r)} |f'_i(y_i)|. \end{aligned} \quad (10)$$

We note that we have similar bounds for our sample standard deviation estimates, as  $\hat{\sigma}_i^{(r,L)} \triangleq (\hat{\sigma}_i^{(r)} - C_\sigma(i, r))_+$  and  $\hat{\sigma}_i^{(r,U)} = \hat{\sigma}_i^{(r)} + C_\sigma(i, r)$ . With these in hand we are able to define Algorithm 3, which operates in a manner similar to Algorithm 1. It proceeds in rounds of increasing accuracy, first coarsely estimating all the means and standard deviations via uniform sampling then refining estimates on higher variance arms by sampling proportionally to a (pessimistic) estimate of their standard deviation. At the end of each round termination sampling allocations  $\tilde{T}_i^{(r)}$  are constructed, and the algorithm samples according to these allocations and terminates if the following termination condition is met:

$$\sum_{i=1}^n \frac{(\hat{g}_i^{(r,U)} \hat{\sigma}_i^{(r,U)})^2}{\tilde{T}_i^{(r)}} + \frac{\varepsilon}{3} \max_i \frac{\hat{g}_i^{(r,U)}}{\tilde{T}_i^{(r)}} \leq \frac{\varepsilon^2}{8 \log(8/\delta)}. \quad (11)$$

---

**Algorithm 3** Adaptive function approximation under heteroskedastic noise

---

- 1: **Input:** arms  $[n]$ , function  $f$ , target accuracy  $\varepsilon$ , error probability  $\delta$
  - 2: Set  $B_0 = \frac{2n^2 L \log(8n/\delta)}{\varepsilon}$
  - 3: **for**  $r = 1, 2, \dots$  **do**
  - 4:   Set round budget  $B_r = B_0 2^r$
  - 5:   Pull each arm  $\lceil B_r/n \rceil$  times
  - 6:   Compute  $\hat{\boldsymbol{\mu}}^{(r)}, \hat{\boldsymbol{\sigma}}^{(r)}$
  - 7:   Construct  $\{\hat{g}_i^{(r,L)}, \hat{g}_i^{(r,U)}\}$  as in (10)
  - 8:   Compute sampling frequencies  $\beta_i^{(r)} \propto \hat{\sigma}_i^{(r,L)}$
  - 9:   Pull arm  $i$   $T_i^{(r)} = \lceil (\beta_i^{(r)} + 1/n) B_r \rceil$  times
  - 10:   Update  $\hat{\boldsymbol{\mu}}^{(r)}, \hat{\boldsymbol{\sigma}}^{(r)}$ , recompute  $\{\hat{g}_i^{(r,L)}, \hat{g}_i^{(r,U)}\}$
  - 11:   Compute frequencies  $\alpha_i^{(r)} \propto \hat{g}_i^{(r,L)} \hat{\sigma}_i^{(r,L)}$
  - 12:   Set  $\tilde{T}_i^{(r)} \triangleq \lceil (\alpha_i + \frac{1}{n}) B_r \rceil$
  - 13:   **if** Termination condition in (11) is met **then**
  - 14:     Pull arm  $i$   $\tilde{T}_i^{(r)}$  times to construct  $\tilde{\boldsymbol{\mu}}$
  - 15:     **return**  $f(\tilde{\boldsymbol{\mu}})$
  - 16:   **end if**
  - 17: **end for**
- 

We are able to state the following Theorem regarding the performance of Algorithm 3, expressing our sample complexity in big Oh notation with respect to  $\varepsilon$  for clarity, but detailing all the lower order dependencies on  $\{g_i\}, \{\sigma_i\}, n, \varepsilon$ , and  $\delta$  with explicit constants in Appendix D (Lemma 11).

**Theorem 3.** *Algorithm 3 successfully outputs  $f(\tilde{\boldsymbol{\mu}})$  such that  $|f(\tilde{\boldsymbol{\mu}}) - f(\boldsymbol{\mu})| \leq \varepsilon$  with probability at least  $1 - \delta$  using a number of arm pulls at most*

$$O\left(\frac{(\sum_i g_i \sigma_i)^2 \log(1/\delta)}{\varepsilon^2}\right) + o(\varepsilon^{-2}).$$

We prove Theorem 3 in Appendix D, beginning by showing that the estimators  $\hat{\boldsymbol{\mu}}_i^{(r)}$  and  $\hat{\sigma}_i^{(r)}$  stay within their confidence intervals. This yields sampling frequencies  $\beta_i^{(r)}$  that sample high variance arms sufficiently, which in turn results in  $\alpha_i^{(r)}$  that ensure that arms with large  $g_i$  and  $\sigma_i$  are sampled enough. We then show that once sufficiently many rounds have elapsed the algorithm will terminate and output an  $\varepsilon$  accurate estimate. Additionally, while the sampling frequencies  $\alpha_i^{(r)}$  in Algorithm 3 are constructed with simplicity and asymptotic optimality in mind, we discuss further optimizations in Appendix D to improve lower order terms in  $\varepsilon$ .

For the sake of simplicity, in this section we assumed that our function  $f$  is separable. Our algorithm can be modified to accommodate general  $f$  by adding into our  $\beta_i^{(r)}$  sampling frequencies a term proportional to

$(\hat{\sigma}_i^{(r,L)})^2$  to minimize the maximum confidence interval width, as in [Antos et al., 2010]. While here we handled the case of bounded noise distributions, our results can additionally be extended to the case of additive gaussian noise with unknown noise variances  $\sigma_i^2$ .

### 5.3 Lower bound

As previously discussed, our algorithms appears to be order optimal in terms of all relevant problem parameters in the high accuracy regime. To this end we provide a matching lower bound for the linear setting where  $f(\boldsymbol{\mu}) = \mathbf{g}^\top \boldsymbol{\mu}$ , a plug-in estimator is used, and our arm pulls for arm  $i$  are corrupted by additive Gaussian noise with variance  $\sigma_i^2$ .

**Proposition 1.** *In order to estimate  $f(\boldsymbol{\mu}) = \mathbf{g}^\top \boldsymbol{\mu}$  to accuracy  $\varepsilon$  with error probability at most  $\delta$ , where pulls from arm  $i$  are corrupted by independent additive Gaussian noise with variance  $\sigma_i^2$ , a plug-in estimator based on any static sampling allocation  $\{T_i\}$  where  $\sum_{i=1}^n T_i = T$  requires*

$$T \geq \left(\sum_i |g_i| \sigma_i\right)^2 \varepsilon^{-2} \log(1/4\delta).$$

We conjecture that this lower bound extends beyond linear functions to general  $f$  with  $L$ -Lipschitz gradients. This bound should be expected to hold, as in the high accuracy regime where  $\varepsilon \rightarrow 0$  the function is essentially linear within the region of estimation. Similar techniques can be used to extend this lower bound to static allocations for such  $f$  when  $\varepsilon$  is sufficiently small and  $T_i = \omega(\varepsilon^{-1})$ , as then the error in the linear approximation can be bounded with probability at least  $1 - \delta$  as  $\frac{L}{2} \|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_2^2 \geq \frac{L}{2} \max_i T_i^{-1} \log(1/\delta) = o(\varepsilon)$ . The technical difficulty in providing an information theoretic lower bound in this more general setting is that the number of pulls per arm cannot be independently bounded, as the estimation error accumulates jointly across the coordinates. This lower bound, proved in Appendix E, shows that our scheme's dependence on all relevant problem parameters  $\{g_i\}, \{\sigma_i\}, \varepsilon$ , and  $\delta$ , is order optimal in the linear case.

## 6 Numerical Experiments

We corroborate our theoretical claims with numerical simulations to show the practical utility and desirable finite sample performance of our scheme. Note that, while not shown in these plots, one significant advantage of our schemes over optimism-based ones is that our schemes offer valuable fixed-confidence stopping conditions. Additional experimental details can be found in Appendix F.

## 6.1 Estimating $\|Ax\|_2^2$

Here we tackle the problem of estimating the power of a matrix vector product, which arises in statistical testing and Fourier signal processing applications. In our simulations we compare our Algorithm 1 with a uniform sampling benchmark, which pulls all arms  $T/n$  times for a total budget of  $T$ . Due to the non-linearity of  $f(\boldsymbol{\mu}) = \|\boldsymbol{\mu}\|_2^2$ , we cannot compare with the software reliability estimation algorithms or the optimism-based regret minimization ones. We are able to construct a modified optimism-based algorithm for this setting, which we discuss and show our improvement over in Appendix F.1.

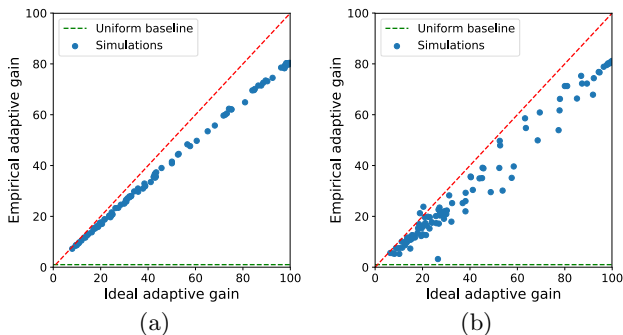


Figure 1: Time until termination for (a)  $f(\boldsymbol{\mu}) = \|\boldsymbol{\mu}\|_2^2$  and (b)  $f(\boldsymbol{\mu}) = \mathbf{g}^\top \boldsymbol{\mu}$  for  $\varepsilon = 10^{-4}$ ,  $\delta = .01$ . Dashed red line indicates theoretically predicted gain from adaptivity, the line  $y = x$ . 100 randomly generated problem instances are shown.

In Figure 1 we plot the improvement in stopping time afforded by adaptivity. Taking the ratio of the number of samples required by the uniform sampling based algorithm to terminate to the number of samples required by our adaptive sampling based algorithm to terminate, we see that these ratios follow a linear trend. In these plots we took  $\tilde{T}_i^{(r)} = \lceil 10\alpha_i B_r \rceil$ , which does not orderwise impact our theoretical guarantees but allows for superior finite sample performance. While theory would predict that all points fall on the line  $y = x$ , finite sample issues and our choice of  $\tilde{T}_i^{(r)}$  yield the resulting slope of  $\approx 10/12$ , for which we provide further discussion in Appendix F. Additionally, we reduce the confidence intervals constants to allow for more aggressive sampling. All modifications from the written algorithm are detailed in Appendix F.2.

In Figure 2 we see the dramatic gain afforded by adaptivity in the estimation of  $\|Ax\|_2^2$ . For this random example the gain between uniform and adaptive sampling was projected to be 60.5. While in these experiments we utilize  $A, x$  such that the variance of our drawn samples is approximately 1 for simplicity, in many applications one would need to adapt to the unknown variance of different entries. In order to accomplish this, a bound on  $\max_{i,j} |A_{ij}x_j|$  is required.

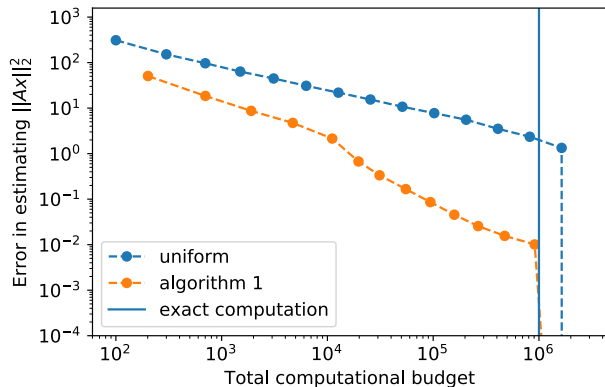


Figure 2: Estimation error of  $\|Ax\|_2^2$  for matrix  $A$ , vector  $x$ , for  $n = 100$  and  $d = 10k$ .

This allows for adaptation to the unknown variances  $\sigma_i^2 = d \sum_{j=1}^d (A_{ij} - \mu_i)^2$ .

Note that in this computational setting, we have that in addition to high probability bounds on the sample complexity of the algorithm, we can provide bounds on the expected runtime as well, as the total number of samples required is upper bounded by  $nd$  with probability 1.

## 6.2 Linear $f$ with heteroskedastic noise

We provide further numerical validation of our method by demonstrating its performance in the software reliability estimation setting, where we wish to estimate  $f(\boldsymbol{\mu}) = \mathbf{g}^\top \boldsymbol{\mu}$ , and when we query arm  $i$  we obtain a Bernoulli sample with mean  $\mu_i$  (heteroskedastic noise) indicating whether the test failed. We see that our general algorithm is able to outperform the optimism-based one of [Carpentier et al., 2015] in both error and wall-clock time, while additionally providing stopping time guarantees.

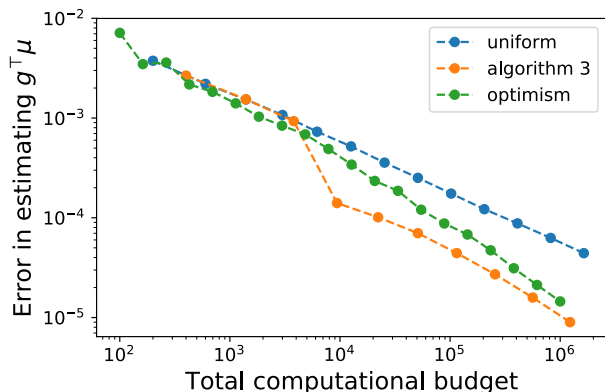


Figure 3: Estimation error of  $\mathbf{g}^\top \boldsymbol{\mu}$  under Bernoulli noise for  $n = 100$ ,  $g_i$  uniform in  $[0,1]$  and  $\boldsymbol{\mu}$  normalized pareto distribution.

In Figure 3 we see the favorable comparison of Algorithm 3 to the optimism based method. Even though



Algorithm 3 is designed with a fixed confidence stopping condition in mind, and does not optimize for this fixed budget metric of error after  $T$  samples, we see that it is in fact able to slightly outperform the optimism-based algorithm. Both adaptive algorithms exhibit an inflection point (which is much sharper in Algorithm 3) when they identify the most important coordinate with largest  $g_i \mu_i$  and are able to use the majority of their samples on it. This effect can also be seen to some degree in Figure 2. Additional numerical experiments can be found in Appendix F.1.

	Uniform	Algorithm 3	Optimism
$T = 10^4$	$3.4 \times 10^{-3}$	$7.0 \times 10^{-3}$	$3.2 \times 10^{-1}$
$T = 10^5$	$5.2 \times 10^{-3}$	$8.4 \times 10^{-3}$	3.1
$T = 10^6$	$7.5 \times 10^{-3}$	$1.1 \times 10^{-2}$	$3.1 \times 10$

Table 1: Wall-clock time comparison for estimation of  $f(\boldsymbol{\mu}) = \mathbf{g}^\top \boldsymbol{\mu}$  as in Figure 3, averaged over 100 trials. Simulation details in Appendix F.

We see that while the statistical performance of the optimism-based algorithm is comparable to our scheme, it is computationally much slower. This is shown in Table 1, where we see that our algorithm is over 3 orders of magnitude faster when run for  $T = 10^6$  samples, and is almost as fast as the batched pure uniform sampling algorithm. The runtimes of the two batched methods increase extremely slowly (logarithmically) as the budget increases, while the fully sequential optimism based algorithm suffers from a runtime linear in  $T$ .

## 7 Conclusions and Future Work

In this paper, we considered the problem of estimating the value of a known smooth function  $f$  at an unknown point  $\boldsymbol{\mu} \in \mathbb{R}^n$ . We provided a novel algorithm to solve this problem, showing that adaptively estimating the important coordinates of the unknown point can yield dramatic performance improvements. We provided a matching lower bound in the linear setting, showing that our algorithm achieves order optimal performance in terms of all relevant problem parameters. We demonstrated the broad applicability of our scheme by showing its extensions to a thresholding decision setting, as well as our algorithms’ ability to adapt to unknown variances.

There are several interesting directions for future work. We showed in the special case where  $f(\boldsymbol{\mu}) = \|\boldsymbol{\mu}\|_2^2$  that additional structure can lead to a more efficient algorithm. One natural question is what other classes of functions can yield improved rates. Additionally, while in this work we only considered bounded noise distributions (or Gaussian noise with unknown variance), one should be able to use more sophisticated estima-

tors like Catoni’s M-estimator [Catoni, 2012] to accommodate more general noise distributions with moments of order  $1 + \varepsilon$  for  $\varepsilon \in (0, 1]$  [Bubeck et al., 2013]. Finally, the technique we use for estimating  $\|Ax\|_2$  can also be utilized for the estimation of  $\|AB\|_F^2$ , as this can be viewed as the  $\ell_2$  norm of the vectorized resulting matrix. This matrix matrix product contains additional structure, and has been studied from alternative perspectives in the Monte Carlo matrix multiplication literature [Drineas et al., 2006], so designing wall-clock efficient algorithms for this setting is an interesting question for future work.

## References

- [Antos et al., 2010] Antos, A., Grover, V., and Szepesvári, C. (2010). Active learning in heteroscedastic noise. *Theoretical Computer Science*, 411(29-30):2712–2728.
- [Bagaria et al., 2021] Bagaria, V., Baharav, T. Z., Kamath, G. M., and David, N. T. (2021). Bandit-based monte carlo optimization for nearest neighbors. *IEEE Journal on Selected Areas in Information Theory*.
- [Bagaria et al., 2018] Bagaria, V., Kamath, G., Ntranos, V., Zhang, M., and Tse, D. (2018). Medoids in almost-linear time via multi-armed bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 500–509. PMLR.
- [Baharav and Tse, 2019] Baharav, T. and Tse, D. (2019). Ultra fast medoid identification via correlated sequential halving. *Advances in Neural Information Processing Systems*, 32.
- [Balkanski and Singer, 2018] Balkanski, E. and Singer, Y. (2018). The adaptive complexity of maximizing a submodular function. In *Proceedings of the 50th annual ACM SIGACT symposium on theory of computing*, pages 1138–1151.
- [Bubeck, 2013] Bubeck, S. (2013). Oracle complexity of smooth convex functions.
- [Bubeck and Cesa-Bianchi, 2012] Bubeck, S. and Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122.
- [Bubeck et al., 2013] Bubeck, S., Cesa-Bianchi, N., and Lugosi, G. (2013). Bandits with heavy tail. *IEEE Transactions on Information Theory*, 59(11):7711–7717.
- [Cai et al., 2004] Cai, K.-Y., Li, Y.-C., and Liu, K. (2004). Optimal and adaptive testing for software

- reliability assessment. *Information and Software Technology*, 46(15):989–1000.
- [Carpentier and Munos, 2012] Carpentier, A. and Munos, R. (2012). Adaptive stratified sampling for monte-carlo integration of differentiable functions. *Advances in neural information processing systems*, 25.
- [Carpentier et al., 2015] Carpentier, A., Munos, R., and Antos, A. (2015). Adaptive strategy for stratified monte carlo sampling. *J. Mach. Learn. Res.*, 16:2231–2271.
- [Catoni, 2012] Catoni, O. (2012). Challenging the empirical mean and empirical variance: a deviation study. In *Annales de l’IHP Probabilités et statistiques*, volume 48, pages 1148–1185.
- [Drineas et al., 2006] Drineas, P., Kannan, R., and Mahoney, M. W. (2006). Fast monte carlo algorithms for matrices i: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1):132–157.
- [Esfandiari et al., 2021] Esfandiari, H., Karbasi, A., and Mirrokni, V. (2021). Adaptivity in adaptive submodularity. In *Conference on Learning Theory*, pages 1823–1846. PMLR.
- [Hillel et al., 2013] Hillel, E., Karnin, Z. S., Koren, T., Lempel, R., and Somekh, O. (2013). Distributed exploration in multi-armed bandits. *Advances in Neural Information Processing Systems*, 26.
- [Hu et al., 2013] Hu, H., Jiang, C.-H., Cai, K.-Y., Wong, W. E., and Mathur, A. P. (2013). Enhancing software reliability estimates using modified adaptive testing. *Information and Software Technology*, 55(2):288–300.
- [Jamieson et al., 2014] Jamieson, K., Malloy, M., Nowak, R., and Bubeck, S. (2014). lil’ucb: An optimal exploration algorithm for multi-armed bandits. In *Conference on Learning Theory*, pages 423–439.
- [Jamieson and Nowak, 2014] Jamieson, K. and Nowak, R. (2014). Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *Information Sciences and Systems (CISS), 2014 48th Annual Conference on*, pages 1–6. IEEE.
- [Kamath et al., 2020] Kamath, G., Baharav, T., and Shomorony, I. (2020). Adaptive learning of rank-one models for efficient pairwise sequence alignment. *Advances in Neural Information Processing Systems*, 33:7513–7525.
- [Karbasi et al., 2021] Karbasi, A., Mirrokni, V., and Shadravan, M. (2021). Parallelizing thompson sampling. *Advances in Neural Information Processing Systems*, 34.
- [Karnin et al., 2013] Karnin, Z., Koren, T., and Somekh, O. (2013). Almost optimal exploration in multi-armed bandits. In *International Conference on Machine Learning*, pages 1238–1246. PMLR.
- [Karpov and Zhang, 2020] Karpov, N. and Zhang, Q. (2020). Batched coarse ranking in multi-armed bandits. *Advances in Neural Information Processing Systems*, 33.
- [Kaufmann et al., 2016] Kaufmann, E., Cappé, O., and Garivier, A. (2016). On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1):1–42.
- [Kim and Nelson, 2001] Kim, S.-H. and Nelson, B. L. (2001). A fully sequential procedure for indifference-zone selection in simulation. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 11(3):251–273.
- [Kocsis and Szepesvári, 2006] Kocsis, L. and Szepesvári, C. (2006). Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer.
- [Lai and Robbins, 1985] Lai, T. L. and Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22.
- [Lattimore and Szepesvári, 2020] Lattimore, T. and Szepesvári, C. (2020). *Bandit algorithms*. Cambridge University Press.
- [LeJeune et al., 2019] LeJeune, D., Heckel, R., and Baraniuk, R. (2019). Adaptive estimation for approximate  $k$ -nearest-neighbor computations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3099–3107. PMLR.
- [Li et al., 2017] Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. (2017). Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816.
- [Lohr, 2019] Lohr, S. L. (2019). *Sampling: design and analysis*. Chapman and Hall/CRC.
- [Lv et al., 2014] Lv, J., Yin, B.-B., and Cai, K.-Y. (2014). On the asymptotic behavior of adaptive testing strategy for software reliability assessment. *IEEE transactions on Software Engineering*, 40(4):396–412.

- [Mason et al., 2019] Mason, B., Tripathy, A., and Nowak, R. (2019). Learning nearest neighbor graphs from noisy distance samples. *Advances in Neural Information Processing Systems*, 32.
- [Mason et al., 2021] Mason, B., Tripathy, A., and Nowak, R. (2021). Nearest neighbor search under uncertainty. In *Uncertainty in Artificial Intelligence*, pages 1777–1786. Proceedings of Machine Learning Research.
- [Maurer and Pontil, 2009] Maurer, A. and Pontil, M. (2009). Empirical bernstein bounds and sample variance penalization. *Conference on Learning Theory*, 22.
- [Paulson, 1964] Paulson, E. (1964). A sequential procedure for selecting the population with the largest mean from k normal populations. *The Annals of Mathematical Statistics*, pages 174–180.
- [Robbins, 1952] Robbins, H. (1952). Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535.
- [Singhal et al., 2021] Singhal, A., Pirojiwala, S., and Karamchandani, N. (2021). Query complexity of k-nn based mode estimation. In *2020 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE.
- [Sion, 1958] Sion, M. (1958). On general minimax theorems. *Pacific Journal of mathematics*, 8(1):171–176.
- [Tiwari et al., 2020] Tiwari, M., Zhang, M. J., Mayclin, J., Thrun, S., Piech, C., and Shomorony, I. (2020). Banditpam: Almost linear time k-medoids clustering via multi-armed bandits. *Advances in Neural Information Processing Systems*, 33:10211–10222.
- [Wainwright, 2019] Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press.
- [Zhang et al., 2019] Zhang, M., Zou, J., and Tse, D. (2019). Adaptive monte carlo multiple testing via multi-armed bandits. In *International Conference on Machine Learning*, pages 7512–7522. PMLR.

## A $\ell_p$ norms

In this Appendix we analyze our algorithms' performance for  $\ell_p$  norms, where we wish to estimate  $f(\boldsymbol{\mu}) = f(\boldsymbol{\mu}; p) \triangleq \|\boldsymbol{\mu}\|_p^p$  to accuracy  $\varepsilon$ . We focus on the high accuracy regime for the sake of exposition, taking  $\varepsilon \rightarrow 0$  and ignoring  $o(\varepsilon^{-2})$  terms. For  $\|\boldsymbol{\mu}\|_\infty$ , in the high accuracy regime it is order optimal to first identify an  $\varepsilon/2$  optimal arm  $i^*$  such that  $|\mu_{i^*}| \geq \max_i |\mu_i| - \varepsilon/2$ , and then estimate  $\mu_{i^*}$  to accuracy  $\varepsilon/2$ . This two step procedure has an identification cost of  $\tilde{O}(\sum_{i=1}^n (\Delta_i^{(\varepsilon)})^{-2})$  for  $\Delta_i^{(\varepsilon)} = \max(\Delta_i, \varepsilon)$  [Jamieson and Nowak, 2014], after which the identified arm can be pulled  $\tilde{O}(\varepsilon^{-2})$  times to estimate its mean to accuracy  $\varepsilon/2$ , yielding an overall sample complexity of  $\tilde{O}(\varepsilon^{-2} + \sum_{i=1}^n (\Delta_i^{(\varepsilon)})^{-2}) \rightarrow \tilde{O}(\varepsilon^{-2})$ , assuming that only a constant number of arms attain this maximum. Here,  $\tilde{O}$  hides log factors in  $\text{poly}(n, 1/\delta, 1/\varepsilon, 1/\Delta_i)$ . This is in contrast to a uniform sampling based algorithm, which would estimate the means of all arms to accuracy  $\varepsilon/2$  and then output the largest value, requiring  $\tilde{O}(n\varepsilon^{-2})$  samples. This yields a performance improvement afforded by adaptivity of  $\tilde{O}(n)$  when the best arm is unique. As shown in Theorem 1, our adaptive algorithm requires  $O(\|\nabla f(\boldsymbol{\mu})\|_1^2 \varepsilon^{-2} \log(1/\delta))$  samples for smooth  $f$  ignoring lower order terms in  $\varepsilon$ , as opposed to a uniform sampling algorithm which will require  $O(n\varepsilon^{-2} \|\nabla f(\boldsymbol{\mu})\|_2^2 \log(1/\delta))$ , yielding a ratio of  $O(n\|\nabla f(\boldsymbol{\mu})\|_2^2 / \|\nabla f(\boldsymbol{\mu})\|_1^2)$ . On the other end of the spectrum, the case of  $f(\boldsymbol{\mu}) = \|\boldsymbol{\mu}\|_1$  is surprising in that adaptivity does not save samples, as for all  $i$  such that  $\mu_i \neq 0$  we have that  $|\nabla_i f(\boldsymbol{\mu})| = 1$ . Thus, all coordinates are equally important, and it is optimal to uniformly sample the entries. Revisiting  $f(\boldsymbol{\mu}) = \|\boldsymbol{\mu}\|_2^2$ , we have that  $\nabla f(\boldsymbol{\mu}) = 2\boldsymbol{\mu}$ , and so the gain from adaptivity will be  $O(n\|\boldsymbol{\mu}\|_2^2 / \|\boldsymbol{\mu}\|_1^2)$ , potentially a factor of  $n$  improvement. Defining the gain afforded by adaptivity for a fixed vector  $\boldsymbol{\mu}$  as

$$\text{gain}(\|\cdot\|_p^p, \boldsymbol{\mu}) \triangleq \frac{n\|\nabla f(\boldsymbol{\mu}; p)\|_2^2}{\|\nabla f(\boldsymbol{\mu}; p)\|_1^2}, \quad (12)$$

we are able to state the following proposition showing that the gain increases with  $p$  as the function becomes less smooth.

**Proposition 2.** *For  $p \in [1, \infty)$ , the sample complexity improvement  $\text{gain}(\|\cdot\|_p^p, \boldsymbol{\mu})$  in (12) is a monotone nondecreasing function of  $p$ , and a monotone increasing function if the  $\mu_i$  are not all identical.*

*Proof of Proposition 2.* Analyzing our gain, we see that we can assume without loss of generality that  $\mu_i \geq 0$  for all  $i$ . Defining  $h(\boldsymbol{\mu}; p) \triangleq \sum_{i=1}^n \mu_i^{2(p-1)}$  and  $g(\boldsymbol{\mu}; p) \triangleq (\sum_{i=1}^n \mu_i^{p-1})^2$ , we have that

$$\text{gain}(\|\cdot\|_p^p, \boldsymbol{\mu}) = \frac{n\|\nabla f(\boldsymbol{\mu}; p)\|_2^2}{\|\nabla f(\boldsymbol{\mu}; p)\|_1^2} = \frac{n \sum_{i=1}^n \mu_i^{2(p-1)}}{\left(\sum_{i=1}^n \mu_i^{p-1}\right)^2} = \frac{nh(\boldsymbol{\mu}; p)}{g(\boldsymbol{\mu}; p)}. \quad (13)$$

To show that this is a nondecreasing function of  $p$ , we simply need to show that  $h'(\boldsymbol{\mu}; p)g(\boldsymbol{\mu}; p) - h(\boldsymbol{\mu}; p)g'(\boldsymbol{\mu}; p) \geq 0$ , where the derivatives are taken with respect to  $p$  for a fixed  $\boldsymbol{\mu}$ . Since coordinates where  $\mu_i = 0$  affect neither the numerator nor the denominator we can remove them from consideration, and assume without loss of generality that  $\mu_i > 0$  for all  $i$ . Taking the derivatives and simplifying, we have that our gain is a non decreasing function of  $p$  as

$$\begin{aligned} & \left(\sum \mu_i^{2(p-1)} \ln \mu_i\right) \left(\sum \mu_i^{p-1}\right) - \left(\sum \mu_i^{2(p-1)}\right) \left(\sum \mu_i^{p-1} \ln \mu_i\right) \\ &= \left(\sum_{i,j} \mu_i^{2(p-1)} \mu_j^{p-1} \ln \mu_i\right) - \left(\sum_{i,j} \mu_i^{2(p-1)} \mu_j^{p-1} \ln \mu_j\right) \\ &\stackrel{(a)}{=} \sum_{i < j} \left(\left(\mu_i^{2(p-1)} \mu_j^{p-1} \ln \mu_i + \mu_j^{2(p-1)} \mu_i^{p-1} \ln \mu_j\right) - \left(\mu_i^{2(p-1)} \mu_j^{p-1} \ln \mu_j + \mu_j^{2(p-1)} \mu_i^{p-1} \ln \mu_i\right)\right) \\ &= \sum_{i < j} \left(\mu_i^{2(p-1)} \mu_j^{p-1} \ln(\mu_i/\mu_j) + \mu_j^{2(p-1)} \mu_i^{p-1} \ln(\mu_j/\mu_i)\right) \\ &= \sum_{i < j} \left(\mu_i^{p-1} \mu_j^{p-1} \ln(\mu_i/\mu_j) \left(\mu_i^{p-1} - \mu_j^{p-1}\right)\right) \\ &\geq 0. \end{aligned} \quad (14)$$

Where in (a) we grouped pairs of terms  $(i, j)$  and  $(j, i)$  from both sums, and in the last inequality we observe that for  $\mu_i, \mu_j > 0$ , the expression is always nonnegative. This shows that the derivative of the gain with respect to  $p$  is nonnegative for all  $p \in [1, \infty)$ . Examining this final line more closely, we see that unless  $\mu_i = \mu_j$  for all pairs  $(i, j)$  there will be a positive term in this sum, indicating that the gain is a monotone increasing function of  $p$  when the  $\mu_i$  are not all identical.  $\square$

## B Proofs of Algorithm 1

**Proof sketch:** we define the good event  $\xi$  where the mean estimators  $\{\hat{\mu}_i^{(r)}\}$  stay within their confidence intervals, and show in Lemma 1 that this event occurs with probability at least  $1 - \delta/3$ . This yields control of our gradient bounds  $\{\hat{g}_i^{(r,L)}, \hat{g}_i^{(r,U)}\}$ , and guarantees that the sampling frequencies  $\alpha_i^{(r)}$  are sufficiently large for important arms. Then, we show in Lemma 2 that the error due to higher order terms is less than  $\varepsilon/2$  due to our choice of  $B_0$ . We then analyze our first order error term, and show that this is sufficiently small when the termination condition is met. Finally, we bound the number of rounds required until the termination condition is met.

### B.1 Algorithm terminates correctly

We begin by showing that the confidence intervals on the mean estimators hold. To do this we define the good event  $\xi$ , where our confidence intervals on our mean estimators hold.

$$\xi \triangleq \bigcap_{i \in [n], r \in \mathbb{N}} \left\{ |\hat{\mu}_i^{(r)} - \mu_i| < C_r \right\} \quad (15)$$

This event occurs with very high probability, as formalized in the lemma below.

**Lemma 1.** *The good event  $\xi$  where the estimators  $\{\hat{\mu}_i^{(r)}\}$  stay within their confidence intervals satisfies  $\mathbb{P}(\xi) \geq 1 - \delta/3$ .*

*Proof.* Analyzing  $\xi$ , we have that the probability that any of our confidence intervals fail is at most

$$\mathbb{P}(\bar{\xi}) = \mathbb{P}\left(\exists i \in [n], r \in \mathbb{N}, \text{ s.t. } |\hat{\mu}_i^{(r)} - \mu_i| \geq C_r\right) \leq \sum_{i \in [n], r \in \mathbb{N}} 2 \exp(-\log(12nr^2/\delta)) \leq \delta/3. \quad (16)$$

$\square$

The rest of the analysis will be conditioned on this good event  $\xi$ . Analyzing the gradient estimators  $\{\hat{g}_i^{(r,L)}, \hat{g}_i^{(r,U)}\}$ , we have for all  $i, r$  that

$$g_i + 2L\sqrt{n}C_r \geq \hat{g}_i^{(r,U)} \geq g_i \geq \hat{g}_i^{(r,L)} \geq g_i - 2L\sqrt{n}C_r \quad (17)$$

This is because, conditioned on  $\xi$ ,

$$\begin{aligned} \hat{g}_i^{(r,L)} &= \min_{\mathbf{y}: \|\mathbf{y} - \hat{\boldsymbol{\mu}}^{(r)}\|_\infty \leq C_r} |\nabla_i f(\mathbf{y})| \\ &\geq g_i - \max_{\mathbf{y}: \|\mathbf{y} - \boldsymbol{\mu}\|_\infty \leq 2C_r} |\nabla_i f(\mathbf{y}) - \nabla_i f(\boldsymbol{\mu})| \\ &\geq g_i - \max_{\mathbf{y}: \|\mathbf{y} - \boldsymbol{\mu}\|_\infty \leq 2C_r} \|\nabla f(\mathbf{y}) - \nabla f(\boldsymbol{\mu})\| \\ &\geq g_i - 2L\sqrt{n}C_r. \end{aligned} \quad (18)$$

Additionally,  $g_i \geq \hat{g}_i^{(r,L)}$  as the set we are minimizing over includes  $\mathbf{y} = \boldsymbol{\mu}$ . The result for  $\hat{g}_i^{(r,U)}$  follows identically.

With Lemma 1 in hand, we can now proceed with the analysis knowing that our gradient estimators stay within their confidence intervals for the duration of the algorithm with high probability. This result directly implies the

accuracy of our sampling frequencies  $\alpha_i^{(r)}$ , where we can see that after defining our problem complexity measure  $H \triangleq \sum_i g_i = \|\nabla f(\boldsymbol{\mu})\|_1$  we have that

$$\alpha_i^{(r)} = \frac{\hat{g}_i^{(r,L)}}{\sum_j \hat{g}_j^{(r,L)}} \geq \begin{cases} \frac{g_i}{2H} & \text{if } g_i \geq 4L\sqrt{n}C_r, \\ 0 & \text{otherwise.} \end{cases} \quad (19)$$

We now analyze the higher order error terms in our output  $f(\tilde{\boldsymbol{\mu}})$  caused by not knowing the true gradient.

**Lemma 2.** *If each arm is sampled at least  $2nL \log(6n/\delta)\varepsilon^{-1}$  times, then with probability at least  $1 - \delta/3$*

$$\frac{L}{2} \|\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_2^2 \leq \varepsilon/2$$

*Proof.* We have by Hoeffding's inequality [Wainwright, 2019] that if arm  $i$  has been pulled  $\tilde{T}_i^{(r)}$  times then

$$\mathbb{P}\left(\frac{L}{2} \|\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_2^2 \geq \varepsilon/2\right) \leq \mathbb{P}\left(\|\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_\infty \geq \frac{\varepsilon}{nL}\right) \leq 2 \sum_{i=1}^n \exp\left(-\frac{\tilde{T}_i^{(r)} \varepsilon}{2nL}\right) \leq \delta/3, \quad (20)$$

where the last inequality holds for  $\tilde{T}_i^{(r)} \geq 2nL \log(6n/\delta)\varepsilon^{-1}$  which is satisfied by  $B_0 \geq 2n^2L \log(6n/\delta)\varepsilon^{-1}$ .  $\square$

This requirement is non-adaptive, being independent of  $\boldsymbol{\mu}$ , where above this sampling threshold these higher order effects can be readily bounded.

Moving to the error in our linear approximation in (2), we can design our ideal termination condition by appealing to the linear case, noting that by Hoeffding's inequality

$$\sum_{i=1}^n \frac{g_i^2}{\tilde{T}_i^{(r)}} \leq \frac{\varepsilon^2}{8 \log(6/\delta)} \implies \mathbb{P}\left(|\nabla f(\boldsymbol{\mu})^\top(\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu})| > \varepsilon/2\right) \leq \delta/3. \quad (21)$$

Without knowledge of the true  $\nabla f(\boldsymbol{\mu})$  however, we cannot compute this stopping condition. This necessitates an algorithmic stopping condition involving our estimators  $\hat{g}_i^{(r,L)}$ . Conditioning on the good event  $\xi$ , we proceed with our analysis in this section assuming that our gradient bounds are well behaved. This leads to the stopping condition in Algorithm 1 of

$$\sum_{i=1}^n \frac{(\hat{g}_i^{(r,U)})^2}{\tilde{T}_i^{(r)}} \leq \frac{\varepsilon^2}{8 \log(6/\delta)} \implies \sum_{i=1}^n \frac{g_i^2}{\tilde{T}_i^{(r)}} \leq \frac{\varepsilon^2}{8 \log(6/\delta)}, \quad (22)$$

where the implication holds on  $\xi$ , meaning that our algorithm will not incorrectly terminate early except on this error event.

## B.2 Bounding the sample complexity

We analyze the number of pulls required for the bound in (22) to hold by splitting our arms into 2 sets; those with  $g_i$  larger or smaller than  $\tau = 4L\sqrt{n}C_r$ . Note that  $\tilde{B}_r \geq B_r/n$ . Conditioned on  $\xi$  we have that

$$\begin{aligned} \sum_{i=1}^n \frac{(\hat{g}_i^{(r,U)})^2}{\tilde{T}_i^{(r)}} &\stackrel{(a)}{\leq} \sum_{i=1}^n \frac{(g_i + 2L\sqrt{n}C_r)^2}{\tilde{T}_i^{(r)}} \\ &\leq \sum_{i: g_i < \tau} \frac{(g_i + 2L\sqrt{n}C_r)^2}{B_r/n} + \sum_{i: g_i \geq \tau} \frac{(g_i + 2L\sqrt{n}C_r)^2}{\alpha_i B_r} \\ &\leq \sum_{i: g_i < \tau} \frac{36nL^2C_r^2}{B_r/n} + \sum_{i: g_i \geq \tau} \frac{9g_iH}{4B_r} \\ &\leq \frac{36n^3L^2C_r^2}{B_r} + \frac{9H^2}{4B_r}, \end{aligned} \quad (23)$$

where (a) utilizes the event  $\xi$ .

Analyzing the first term in (23), we see that this is less than  $\frac{\varepsilon^2}{16 \log(6/\delta)}$  for all  $r > 0$  as long as  $B_0 \geq 17n^2 L \log(12n/\delta) \varepsilon^{-1}$ , since

$$\frac{36n^3 L^2 C_r^2}{B_r} \leq \frac{36n^3 L^2 C_1^2}{2B_0} \leq \frac{18n^4 L^2 \log(12n/\delta)}{B_0^2} \leq \frac{\varepsilon^2}{16 \log(6/\delta)}. \quad (24)$$

Moving to the second term in (23), we see that it is less than  $\frac{\varepsilon^2}{16 \log(6/\delta)}$  when  $B_r \geq 36H^2 \log(6/\delta) \varepsilon^{-2}$ . This means that our algorithm will terminate no later than the first round  $r$  such that

$$B_r \geq 36H^2 \log(6/\delta) \varepsilon^{-2}. \quad (25)$$

Due to the doubling of budgets round to round, the round where this condition is met must satisfy  $B_r \leq 72H^2 \log(6/\delta) \varepsilon^{-2}$ .

Analyzing the total sample complexity, we see that in round  $r$  Algorithm 1 takes at most  $B_r + n = B_0 2^r + n$  samples, yielding a cumulative sample complexity of at most  $2B_r - B_0 + rn$  samples by the end of round  $r$ . After the termination condition is met, the algorithm will perform  $\sum_{i=1}^n \tilde{T}_i^{(r)} \leq 2B_r + n$  additional pulls before returning its estimate. Hence, if our algorithm terminates at the end of round  $r$  its sample complexity can be upper bounded by  $4B_r - B_0 + (r+1)n \leq 5B_r$ , as  $B_0 \geq n$  and  $B_r \geq 2^r n \geq rn$  for  $r \geq 1$ .

Analyzing the failure probability of Algorithm 3, we see that we will output an estimate  $f(\tilde{\boldsymbol{\mu}})$  with error at most  $\varepsilon$  with probability at least  $1 - \delta$ , as by Lemma 1, Lemma 2, and (21) we have that

$$\begin{aligned} \mathbb{P}(|f(\tilde{\boldsymbol{\mu}}) - f(\boldsymbol{\mu})| \geq \varepsilon) &\leq \mathbb{P}(|f(\tilde{\boldsymbol{\mu}}) - f(\boldsymbol{\mu})| \geq \varepsilon \mid \xi) + \mathbb{P}(\bar{\xi}) \\ &\leq \mathbb{P}(|\nabla f(\tilde{\boldsymbol{\mu}})^\top (\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu})| \geq \varepsilon/2 \mid \xi) + \mathbb{P}\left(\frac{L}{2} \|\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_2^2 \geq \varepsilon/2 \mid \xi\right) + \mathbb{P}(\bar{\xi}) \\ &\leq \delta. \end{aligned} \quad (26)$$

This means that Algorithm 1 will with probability at least  $1 - \delta$  return an estimate  $f(\tilde{\boldsymbol{\mu}})$  of  $f(\boldsymbol{\mu})$  such that  $|f(\tilde{\boldsymbol{\mu}}) - f(\boldsymbol{\mu})| \leq \varepsilon$  using at most

$$\frac{85n^2 L \log(12n/\delta)}{\varepsilon} + \frac{360H^2 \log(6/\delta)}{\varepsilon^2} \quad (27)$$

samples, as desired.

**More refined termination conditions:** Note that our termination condition essentially only requires that we provide an upper bound on the first order error. In certain cases, we can utilize the structure of the function to provide tighter, but more computationally intensive, stopping conditions, where on  $\xi$  we have that

$$\sum_{i=1}^n \frac{g_i^2}{\tilde{T}_i^{(r)}} \leq \max_{\mathbf{y}: \|\mathbf{y} - \hat{\boldsymbol{\mu}}^{(r)}\|_\infty \leq C_r} \sum_{i=1}^n \frac{|\nabla_i f(\mathbf{y})|^2}{\tilde{T}_i^{(r)}} \leq \sum_{i=1}^n \frac{\max_{\mathbf{y}: \|\mathbf{y} - \hat{\boldsymbol{\mu}}^{(r)}\|_\infty \leq C_r} |\nabla_i f(\mathbf{y})|^2}{\tilde{T}_i^{(r)}} \leq \sum_{i=1}^n \frac{(|\nabla_i f(\hat{\boldsymbol{\mu}}^{(r)})| + L\sqrt{n}C_r)^2}{\tilde{T}_i^{(r)}}. \quad (28)$$

Currently, our algorithms utilize the rightmost quantity in their stopping conditions, but the middle two can provide better performance depending on the structure of the function. Concretely, the second from right quantity can be thought of as computing “pessimistic” estimators for the magnitudes of the gradient entries, where each gradient entry is assumed to be as large as it could possibly be within its confidence interval. This contrasts with our sampling frequency construction, where for the  $\boldsymbol{\alpha}^{(r)}$  we utilize “optimistic” estimators of our gradient magnitudes, where they are assumed to be as small as they could possibly be. Note that a similar argument can be made for the construction of sampling frequencies, taking

$$\alpha_i^{(r)} = \min_{\mathbf{y}: \|\mathbf{y} - \hat{\boldsymbol{\mu}}^{(r)}\|_\infty} \frac{\nabla_i f(\mathbf{y})}{\|\nabla f(\mathbf{y})\|_1}. \quad (29)$$

This duality between our sampling and termination frequencies allows our algorithm to efficiently run and guarantee an  $\varepsilon$  correct output.

### B.3 Proof of Corollary 1, tightened results for $f(\boldsymbol{\mu}) = \|\boldsymbol{\mu}\|_2^2$

In the case where the function  $f$  is separable, i.e.  $f(\boldsymbol{\mu}) = \sum_{i=1}^n f_i(\mu_i)$ , where each  $f_i$  has  $L$ -Lipschitz derivatives, we are able to give a tighter bound on  $\hat{g}_i^{(r,L)}$  as  $\hat{g}_i^{(r,L)} \geq g_i - 2LC_r$ , since the error in estimating  $\nabla_i f(\boldsymbol{\mu})$  depends only on the error in estimating  $\mu_i$ . This allows us to loosen the termination condition in Line 1 of Algorithm 1, and provide a tighter bound on the lower order sample complexity terms. If in addition  $f_i(\mu_i) = \mu_i^2$  (more generally, quadratic), then we are able to partially correct for the second order error, as our plug-in estimator yields  $f_i(\hat{\mu}_i) = \mu_i^2 + 2\mu_i W_i / \sqrt{\tilde{B}_r} + W_i^2 / \tilde{B}_r$  for standard normal  $W_i$ . While the crossterm has zero mean and involves the unknown  $\mu_i$ , our standard first order error term, the final term  $W_i^2 / \tilde{B}_r$  has expectation  $1 / \tilde{B}_r$  which does not depend on any unknown parameters, and can be subtracted to yield an *unbiased* estimator of  $f(\boldsymbol{\mu})$ . Dropping the tilde and overloading  $T_i$  as  $T_i = \tilde{T}_i^{(r)}$  for notational simplicity, we see that estimating  $f_i(\mu_i)$  as  $f_i(\hat{\mu}_i^{(r)}) - \frac{1}{T_i}$  allows us to attain a tighter bound, as stated in Corollary 1.

*Proof of Corollary 1.* Expanding the error of our approximation for our modified algorithm, we have that

$$\begin{aligned} \mathbb{P} \left( \left| \|\hat{\boldsymbol{\mu}}\|_2^2 - \sum_{i=1}^n \frac{1}{T_i} - \|\boldsymbol{\mu}\|_2^2 \right| \geq \varepsilon \right) &= \mathbb{P} \left( \left| \sum_{i=1}^n \left( \mu_i + W_i / \sqrt{T_i} \right)^2 - \frac{1}{T_i} - \sum_{i=1}^n \mu_i^2 \right| \geq \varepsilon \right) \\ &= \mathbb{P} \left( \left| \sum_{i=1}^n \frac{2\mu_i W_i}{\sqrt{T_i}} + \sum_{i=1}^n \frac{1}{T_i} (W_i^2 - 1) \right| \geq \varepsilon \right) \\ &\leq \mathbb{P} \left( \left| \sum_{i=1}^n \frac{2\mu_i W_i}{\sqrt{T_i}} \right| \geq \varepsilon/2 \right) + \mathbb{P} \left( \left| \sum_{i=1}^n \frac{1}{T_i} (W_i^2 - 1) \right| \geq \varepsilon/2 \right). \end{aligned} \quad (30)$$

For each  $i$ ,  $\frac{1}{T_i} (W_i^2 - 1)$  is independent across  $i$ , has 0 mean, and is  $\left(\frac{2}{T_i}, \frac{4}{T_i}\right)$ -sub-exponential [Wainwright, 2019].

This shows that the sum is  $(\nu_*, b_*)$ -sub-exponential, where  $\nu_* = 2\sqrt{\sum_{i=1}^n T_i^{-2}}$  and  $b_* = \frac{4}{\min_i T_i}$ . Hence,

$$\mathbb{P} \left( \left| \sum_{i=1}^n \frac{Z_i^2}{T_i} - \frac{1}{T_i} \right| > \varepsilon/2 \right) \leq \begin{cases} 2 \exp\left(-\frac{\varepsilon^2}{8 \sum_{i=1}^n T_i^{-2}}\right) & \text{if } 0 \leq \varepsilon \leq \frac{\nu_*}{b_*} \\ 2 \exp\left(-\frac{\varepsilon \min_i T_i}{8}\right) & \text{if } \varepsilon > \frac{\nu_*}{b_*} \end{cases} \quad (31)$$

In order to guarantee that our higher order error terms are less than  $\varepsilon/2$  with probability at least  $1 - \delta/3$ , we tackle the two possible cases. In the first case  $\sum_i T_i^{-2} \leq 4n^3/B_r^2 \leq n^3/B_0^2$  and so  $B_0 \geq \sqrt{8n^{3/2} \log(6/\delta)} \varepsilon^{-1}$  is sufficient. In the second case,  $\min_i T_i \geq B_r/n \geq 2B_0/n$ , and so  $B_0 \geq 4n \log(6/\delta) \varepsilon^{-1}$  is sufficient. Thus, taking  $B_0 = 4n^{3/2} \log(6/\delta) \varepsilon^{-1}$  guarantees that our error due to these second order terms in (31) is less than  $\varepsilon/2$  with probability at least  $1 - \delta/3$ .

With our tighter bound on  $g_i$  due to the separability of  $f$  across coordinates, we have that the first term in (23), due to indices with  $g_i < \tau$ , is upper bounded by  $36n^2 L^2 C_r^2 / B_r$ . This can be made sufficiently small by taking  $B_0 \geq 17n^{3/2} L \log(12n/\delta) \varepsilon^{-1}$ .

Combining these improved bounds on  $B_0$  with the analysis from Theorem 1 yields a reduced overall sample complexity of

$$\frac{85n^{3/2} L \log(12n/\delta)}{\varepsilon} + \frac{360H^2 \log(6/\delta)}{\varepsilon^2} \quad (32)$$

analogously to (27).  $\square$

## C Proofs for Algorithm 2: thresholding

**Proof sketch:** the proof proceeds similarly to that of Theorem 1, with some interesting nuances stemming from the fact that our  $\varepsilon$  is now unknown. Concretely, since we do not a priori know what accuracy to estimate  $f(\boldsymbol{\mu})$  to, we need to sample according to our estimated optimal frequencies  $\alpha_i^{(r)}$  in every round to generate  $f(\tilde{\boldsymbol{\mu}}^{(r)})$  to estimate the necessary accuracy as  $|f(\tilde{\boldsymbol{\mu}}^{(r)}) - \tau|$ . We begin by defining the event  $\xi$  where our estimators  $f(\tilde{\boldsymbol{\mu}}^{(r)})$  fall within their  $C_r^f$  confidence intervals, and show in Lemma 3 that this occurs with probability at least  $1 - 2\delta/3$ .



Next, we show that on  $\xi$  and  $\tilde{\xi}$ , our algorithm will terminate correctly. Then, since our estimators  $f(\tilde{\boldsymbol{\mu}}^{(r)})$  are correct up to an error of  $C_r^f$ , we show that when  $C_r^f < |f(\boldsymbol{\mu}) - \tau|/2$ , the termination condition must be met. Finally, we bound the number of samples required for this to occur.

### C.1 Algorithm terminates correctly

As before we condition on the good event  $\xi$ , where our mean estimators fall within their confidence intervals. We define an additional good event  $\tilde{\xi}$  where our estimators  $f(\tilde{\boldsymbol{\mu}}^{(r)})$  fall within their confidence intervals as

$$\tilde{\xi} \triangleq \bigcap_{r \in \mathbb{N}} \left\{ |f(\tilde{\boldsymbol{\mu}}^{(r)}) - f(\boldsymbol{\mu})| < C_r^f \right\}. \quad (33)$$

With this, we can state the following Lemma.

**Lemma 3.** *Following Algorithm 2, we have that*

$$\mathbb{P}(\tilde{\xi} \mid \xi) \geq 1 - 2\delta/3$$

where  $\xi$  is the event that our confidence intervals on the estimators  $\hat{\boldsymbol{\mu}}_i^{(r)}$  hold.

*Proof.* Due to the  $L$ -Lipschitz gradients of  $f$  we have that

$$|f(\tilde{\boldsymbol{\mu}}) - f(\boldsymbol{\mu})| \leq |\nabla f(\boldsymbol{\mu})^\top (\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu})| + \frac{L}{2} \|\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_2^2, \quad (34)$$

and so we can upper bound the deviation probability as

$$\begin{aligned} & \mathbb{P}\left(|f(\tilde{\boldsymbol{\mu}}^{(r)}) - f(\boldsymbol{\mu})| \geq C_r^f \mid \xi\right) \\ & \leq \mathbb{P}\left(|\nabla f(\boldsymbol{\mu})^\top (\tilde{\boldsymbol{\mu}}^{(r)} - \boldsymbol{\mu})| + \frac{L}{2} \|\tilde{\boldsymbol{\mu}}^{(r)} - \boldsymbol{\mu}\|_2^2 \geq \sqrt{2 \log(24r^2/\delta)} \sum_i \frac{g_i^2}{\tilde{T}_i^{(r)}} + C_{r,2}^f \mid \xi\right) \\ & \leq \mathbb{P}\left(|\nabla f(\boldsymbol{\mu})^\top (\tilde{\boldsymbol{\mu}}^{(r)} - \boldsymbol{\mu})| \geq \sqrt{2 \log(24r^2/\delta)} \sum_i \frac{g_i^2}{\tilde{T}_i^{(r)}} \mid \xi\right) + \mathbb{P}\left(\frac{L}{2} \|\tilde{\boldsymbol{\mu}}^{(r)} - \boldsymbol{\mu}\|_2^2 \geq C_{r,2}^f \mid \xi\right) \\ & \leq 2\mathbb{P}\left(|\nabla f(\boldsymbol{\mu})^\top (\tilde{\boldsymbol{\mu}}^{(r)} - \boldsymbol{\mu})| \geq \sqrt{2 \log(24r^2/\delta)} \sum_i \frac{g_i^2}{\tilde{T}_i^{(r)}}\right) + 2\mathbb{P}\left(\frac{L}{2} \|\tilde{\boldsymbol{\mu}}^{(r)} - \boldsymbol{\mu}\|_2^2 \geq C_{r,2}^f\right). \end{aligned} \quad (35)$$

where we used the fact that on  $\xi$  we have that  $\hat{g}_i^{(r,U)} \geq g_i$ , allowing us to bound the width of  $C_{r,1}^f$  in terms of deterministic quantities, the  $g_i$  (recalling the definitions of these constants in (7)). We additionally noted that for an event  $E$ ,  $\mathbb{P}(E|\xi) = \frac{\mathbb{P}(E) - \mathbb{P}(E|\tilde{\xi})\mathbb{P}(\tilde{\xi})}{\mathbb{P}(\tilde{\xi})} \leq 2\mathbb{P}(E)$ , as  $\mathbb{P}(\tilde{\xi}) \geq 2/3$ .

Analyzing the first term in (35), we see by Hoeffding's inequality [Wainwright, 2019] that

$$\mathbb{P}\left(|\nabla f(\boldsymbol{\mu})^\top (\tilde{\boldsymbol{\mu}}^{(r)} - \boldsymbol{\mu})| \geq \sqrt{2 \log(24r^2/\delta)} \sum_i \frac{g_i^2}{\tilde{T}_i^{(r)}}\right) \leq 2 \exp(-\log(24r^2/\delta)) \leq \frac{\delta}{12r^2}, \quad (36)$$

as  $\nabla f(\boldsymbol{\mu})^\top (\tilde{\boldsymbol{\mu}}^{(r)} - \boldsymbol{\mu}) \sim \mathcal{N}\left(0, \sum_{i=1}^n \frac{g_i^2}{\tilde{T}_i^{(r)}}\right)$ . Analyzing the second term in (35), we have by Hoeffding's inequality that

$$\mathbb{P}\left(\frac{L}{2} \|\tilde{\boldsymbol{\mu}}^{(r)} - \boldsymbol{\mu}\|_2^2 \geq C_{r,2}^f\right) \leq \frac{\delta}{12r^2}. \quad (37)$$

Plugging (36) and (37) into (35), we have that

$$\mathbb{P}\left(|f(\tilde{\boldsymbol{\mu}}^{(r)}) - f(\boldsymbol{\mu})| \geq C_r^f \mid \xi\right) \leq \frac{\delta}{3r^2}, \quad (38)$$

and so

$$\begin{aligned}
 \mathbb{P}\left(\tilde{\xi} \mid \xi\right) &= 1 - \mathbb{P}\left(\bigcup_{r \in \mathbb{N}} \left\{|f(\tilde{\boldsymbol{\mu}}^{(r)}) - f(\boldsymbol{\mu})| < C_r^f\right\} \mid \xi\right) \\
 &\geq 1 - \sum_{r \geq 1} \frac{\delta}{3r^2} \\
 &\geq 1 - 2\delta/3.
 \end{aligned} \tag{39}$$

□

Hence, on the events  $\tilde{\xi}, \xi$ , our estimator  $f(\tilde{\boldsymbol{\mu}}^{(r)})$  is within its confidence interval for all  $r$ . This implies that on these events, Algorithm 2 will never incorrectly output whether  $f(\tilde{\boldsymbol{\mu}}) > \tau$ : assume for the sake of contradiction that  $\xi, \tilde{\xi}$  occur, and that without loss of generality  $f(\boldsymbol{\mu}) > \tau$  but the algorithm outputs  $f(\tilde{\boldsymbol{\mu}}) < \tau$ . Then for all  $r$  we have

$$f\left(\tilde{\boldsymbol{\mu}}^{(r)}\right) - \tau \geq f(\boldsymbol{\mu}) - \tau - C_r^f > -C_r^f. \tag{40}$$

However, the algorithm will only terminate and output  $f(\tilde{\boldsymbol{\mu}}) < \tau$  if  $f(\tilde{\boldsymbol{\mu}}^{(r)}) - \tau < -C_r^f$ , a contradiction. Hence Algorithm 2 will not output an incorrect decision on  $\xi, \tilde{\xi}$ . Bounding this probability, we have that

$$\mathbb{P}(\xi \cap \tilde{\xi}) = \mathbb{P}(\xi)\mathbb{P}(\tilde{\xi} \mid \xi) \geq (1 - \delta/3)(1 - 2\delta/3) \geq 1 - \delta. \tag{41}$$

## C.2 Bounding the sample complexity

We see that this algorithm's sample complexity will not be too large since it will stop sampling when  $2C_r^f < |f(\boldsymbol{\mu}) - \tau|$ , as then we have that

$$|f(\tilde{\boldsymbol{\mu}}^{(r)}) - \tau| \geq |f(\boldsymbol{\mu}) - \tau| - |f(\tilde{\boldsymbol{\mu}}^{(r)}) - f(\boldsymbol{\mu})| \geq C_r^f, \tag{42}$$

using the fact that  $|f(\tilde{\boldsymbol{\mu}}^{(r)}) - f(\boldsymbol{\mu})| \leq C_r^f$  on  $\xi, \tilde{\xi}$ . We can now provide our proof regarding Algorithm 2.

*Proof of Theorem 2.* Algorithm 2 returns the correct answer with probability at least  $1 - \delta$ , as shown in (41).

To provide the desired sample complexity guarantee, we bound the number of rounds required until  $C_r^f < |f(\boldsymbol{\mu}) - \tau|/2$  on  $\xi, \tilde{\xi}$ . This is accomplished similarly to before;

$$\begin{aligned}
 C_r^f &= \sqrt{2 \log(24r^2/\delta) \sum_i \frac{\left(\hat{g}_i^{(r,U)}\right)^2}{\tilde{T}_i^{(r)}}} + C_{r,2}^f \\
 &\stackrel{(a)}{\leq} \sqrt{2 \log(24r^2/\delta) \sum_i \frac{(g_i + 2L\sqrt{n}C_r)^2}{\tilde{T}_i^{(r)}}} + \frac{Ln \log(24nr^2/\delta)}{\tilde{B}_r} \\
 &\stackrel{(b)}{\leq} \sqrt{\frac{2 \log(24r^2/\delta) (36n^3L^2C_r^2 + 9H^2/4)}{B_r}} + \frac{Ln^2 \log(24nr^2/\delta)}{B_r} \\
 &\leq \sqrt{\frac{9H^2 \log(24r^2/\delta)}{2B_r}} + \sqrt{\frac{144n^3L^2 \log^2(24r^2/\delta)}{B_r^2}} + \frac{Ln^2 \log(24nr^2/\delta)}{B_r}
 \end{aligned} \tag{43}$$

where in (a) we utilized the event  $\xi$ , and in (b) we divided the sum into two two sets based on whether  $g_i$  dominates the confidence interval or vice versa. This must be less than  $|f(\boldsymbol{\mu}) - \tau|/2$  when

$$B_r \geq \frac{18 \log(24r^2/\delta)H^2}{(f(\boldsymbol{\mu}) - \tau)^2} + \frac{24Ln^2 \log(24nr^2/\delta)}{|f(\boldsymbol{\mu}) - \tau|}, \tag{44}$$

and so our algorithm will terminate in the first round  $r$  where this condition is met. Since  $B_r = n2^r$ , we have that  $\log(r) = \log \log(B_r/n)$ , yielding that our algorithm terminates when

$$B_r = \tilde{O}\left(\frac{H^2 \log(1/\delta)}{(f(\boldsymbol{\mu}) - \tau)^2} + \frac{n^2L \log(n/\delta)}{|f(\boldsymbol{\mu}) - \tau|}\right), \tag{45}$$

where  $\tilde{O}$  suppresses  $\log \log(\cdot)$  terms in  $\text{poly}(n, \delta, H, (f(\boldsymbol{\mu}) - \tau)^{-1})$ , which can be explicitly obtained by solving in (44). Since Algorithm 2 makes at most  $3B_r + n$  arm pulls per round, we have that its total budget through round  $r$  can be at most  $\sum_{\ell=1}^r (3B_\ell + n) \leq 6B_r + nr \leq 7B_r$ . This means that the total number of samples used will be

$$\tilde{O} \left( \frac{H^2 \log(1/\delta)}{(f(\boldsymbol{\mu}) - \tau)^2} + \frac{n^2 L \log(n/\delta)}{|f(\boldsymbol{\mu}) - \tau|} \right), \quad (46)$$

yielding the desired result.  $\square$

## D Proofs for Algorithm 3: variance adaptive algorithm

**Proof sketch:** we begin by stating concentration results on sample variances and sample means of bounded random variables in Lemmas 4 and 5. We then redefine the good event  $\xi$  for this algorithm as the event where  $\{\hat{\mu}_i^{(r)}\}$  and  $\{\hat{\sigma}_i^{(r)}\}$  stay within their confidence intervals. We show in Lemma 6 that this occurs with probability at least  $1 - \delta/2$ . These immediately yields control of our gradient estimators and bounds  $\{\hat{g}_i^{(r,L)}, \hat{g}_i^{(r,U)}\}$  and the sampling frequencies  $\{\beta_i^{(r)}\}$  and  $\{\alpha_i^{(r)}\}$ . In Lemma 7 we show that the second order error incurred by Algorithm 3 is at most  $\varepsilon/2$ , and in Lemma 8 bound the first order error by  $\varepsilon/2$  when the termination condition is met. These are combined in Lemma 9 to show that the output has error at most  $\varepsilon$ . Lemmas 10 and 11 guarantee that once the round budget is sufficiently large, the termination condition must be met. Finally, we invert this requirement to yield a bound on the total number of pulls required by our algorithm.

### D.1 Algorithm terminates correctly

We begin by controlling the sample variance  $\hat{\sigma}_i^{(r)}$  of each of the  $\hat{\mu}_i^{(r)}$  estimators. For this, we turn to a concentration result regarding the sample variance.

**Lemma 4** (Concentration of  $\hat{\sigma}_i^{(r)}$ , Theorem 10 [Maurer and Pontil, 2009]). *Let  $n \geq 2$  and  $\mathbf{X} = (X_1, \dots, X_n)$  be a vector of independent random variables with values in  $[0,1]$ . Then for  $\delta' > 0$  we have, writing  $\mathbb{E}V_n$  for  $\mathbb{E}_{\mathbf{X}} V_n(\mathbf{X})$ ,*

$$\mathbb{P} \left( \left| \sqrt{V_n(\mathbf{X})} - \sqrt{\mathbb{E}V_n} \right| > \sqrt{\frac{2 \log(2/\delta')}{n-1}} \right) \leq \delta,$$

where the sample variance  $V_n(\mathbf{X}) \triangleq \frac{1}{n(n-1)} \sum_{i < j} (X_i - X_j)^2$ .

This gives us concentration for the sample standard deviation estimators  $\{\hat{\sigma}_i^{(r)}\}$ , for which we have defined our confidence intervals as

$$C_\sigma(i, r) \triangleq \sqrt{\frac{2 \log(8nr^2/\delta)}{T_i^{(r)} - 1}}. \quad (47)$$

Assuming these confidence intervals hold, we then have that

$$\sigma_i + 2C_\sigma(i, r) \geq \hat{\sigma}_i^{(r,U)} \geq \sigma_i \geq \hat{\sigma}_i^{(r,L)} \geq \sigma_i - 2C_\sigma(i, r) \quad (48)$$

for all  $i, r$  with probability at least  $1 - \delta/3$ .

In the following Lemma we obtain sample variance dependent confidence intervals for our mean estimators  $\{\hat{\mu}_i^{(r)}\}$ .

**Lemma 5** (Concentration of  $\hat{\mu}_i^{(r)}$ , Theorem 4 [Maurer and Pontil, 2009]). *Let  $Z, Z_1, \dots, Z_n$  be i.i.d. random variables with values in  $[0,1]$ , and let  $\delta' > 0$ . Then with probability at least  $1 - \delta'$  we have*

$$\left| \mathbb{E}Z - \frac{1}{n} \sum_{i=1}^n Z_i \right| \leq \sqrt{\frac{2V_n(\mathbf{Z}) \log(4/\delta')}{n}} + \frac{7 \log(4/\delta')}{3(n-1)}$$

Using Lemmas 4 and 5 we construct anytime confidence intervals for our mean estimators  $\{\hat{\mu}_i^{(r)}\}$ . In the  $r$ -th round, the confidence intervals are set to individually fail with probability at most  $\delta' = \delta/(8nr^2)$ , so that when

union bounding over all  $i, r$ , we have that with probability at least  $1 - \delta$  our estimators  $\{\hat{\mu}_i^{(r)}\}$  all fall within their  $C_\mu(i, r)$  confidence intervals, where

$$C_\mu(i, r) \triangleq \hat{\sigma}_i^{(r)} \sqrt{\frac{2 \log(32nr^2/\delta)}{T_i^{(r)}}} + \frac{7 \log(32nr^2/\delta)}{3(T_i^{(r)} - 1)}. \quad (49)$$

With these definitions in hand we can state the following good event  $\xi$ , a modification from Theorem 1, where all our confidence intervals hold:

$$\xi \triangleq \bigcap_{i \in [n], r \in \mathbb{N}} \left\{ \left\{ |\hat{\mu}_i^{(r)} - \mu_i| \leq C_\mu(i, r) \right\} \cap \left\{ |\hat{\sigma}_i^{(r)} - \sigma_i| \leq C_\sigma(i, r) \right\} \right\}. \quad (50)$$

This event occurs with very high probability, as formalized in the lemma below.

**Lemma 6.** *The good event  $\xi$  where the mean and standard deviation estimators stay within their confidence intervals satisfies  $\mathbb{P}(\xi) \geq 1 - \delta/2$ .*

*Proof.* The proof follows directly from Lemmas 4 and 5 and a union bound, with

$$\mathbb{P}(\bar{\xi}) \leq \sum_{i, r} \mathbb{P} \left( \left\{ |\hat{\mu}_i^{(r)} - \mu_i| > C_\mu(i, r) \right\} \cup \left\{ |\hat{\sigma}_i^{(r)} - \sigma_i| > C_\sigma(i, r) \right\} \right) \leq \sum_{i, r} \frac{\delta}{4nr^2} \leq \frac{\delta}{2}. \quad (51)$$

□

With this in hand, we can control the bounds of our gradient estimators  $\{\hat{g}_i^{(r,L)}, \hat{g}_i^{(r,U)}\}$  as before, where on  $\xi$  we have that for all  $i, r$

$$g_i + 2LC_\mu(i, r) \geq \hat{g}_i^{(r,U)} \geq g_i \geq \hat{g}_i^{(r,L)} \geq g_i - 2LC_\mu(i, r). \quad (52)$$

This follows as

$$\begin{aligned} \hat{g}_i^{(r,L)} &= \min_{y: |y - \hat{\mu}_i^{(r)}| \leq C_\mu(i, r)} |f'_i(y)| \\ &\geq g_i - \max_{y: |y - x_i| \leq C_\mu(i, r)} |f'_i(y) - f'_i(x_i)| \\ &\geq g_i - 2LC_\mu(i, r) \end{aligned} \quad (53)$$

We additionally have that  $g_i \geq \hat{g}_i^{(r,L)}$ , as on  $\xi$  the set we are minimizing over includes  $y = \mu_i$ .

These estimators are critical in the construction of our sampling frequencies  $\alpha^{(r)}, \beta^{(r)}$ . On this event  $\xi$ , we see that our  $\beta_i^{(r)}$ , which impact  $T_i^{(r)}$ , satisfy

$$\beta_i^{(r)} = \frac{\hat{\sigma}_i^{(r,L)}}{\sum_j \hat{\sigma}_j^{(r,L)}} \geq \begin{cases} \frac{\sigma_i}{2 \sum_j \sigma_j} & \text{if } \sigma_i \geq 2C_\sigma(i, r), \\ 0 & \text{otherwise.} \end{cases} \quad (54)$$

Regarding the frequencies  $\alpha^{(r)}$ , we see that

$$\alpha_i^{(r)} = \frac{\hat{g}_i^{(r,L)} \hat{\sigma}_i^{(r,L)}}{\sum_j \hat{g}_j^{(r,L)} \hat{\sigma}_j^{(r,L)}} + \frac{1}{n} \geq \begin{cases} \frac{g_i \sigma_i}{2 \sum_j g_j \sigma_j} & \text{if } \sigma_i \geq 2C_\sigma(i, r) \text{ and } g_i \geq 4LC_\mu(i, r), \\ 1/n & \text{otherwise.} \end{cases} \quad (55)$$

We now turn to bounding the error in our estimator  $f(\tilde{\mu})$ , which we can write as

$$|f(\tilde{\mu}) - f(\mu)| \leq |\nabla f(\mu)^\top (\tilde{\mu} - \mu)| + \frac{L}{2} \|\tilde{\mu} - \mu\|_2^2. \quad (56)$$

We begin by bounding the second order error in (56), showing that our initial sampling is sufficient to guarantee that it is less than  $\varepsilon/2$  with probability at least  $1 - \delta/4$ .

**Lemma 7.** *After sampling as in Algorithm 3 we have that*

$$\frac{L}{2} \|\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_2^2 \leq \frac{\varepsilon}{2}$$

with probability at least  $1 - \delta/4$ .

*Proof.* Since our arm pulls are 1-sub-Gaussian, by Hoeffding's inequality we have that

$$\mathbb{P}\left(|\tilde{\mu}_i - \mu_i| > \frac{\varepsilon}{nL}\right) \leq 2 \exp\left(-\frac{\tilde{T}_i^{(r)} \varepsilon}{2nL}\right). \quad (57)$$

Since  $\tilde{T}_i^{(r)} \geq \lceil B_0/n \rceil$ , and by construction  $B_0 = \frac{2n^2 L \log(8n/\delta)}{\varepsilon}$ , we have that

$$\mathbb{P}\left(\|\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_2^2 \leq \varepsilon/L\right) \leq \mathbb{P}\left(\bigcup_i \left\{|\tilde{\mu}_i - \mu_i| > \frac{\varepsilon}{nL}\right\}\right) \leq 2 \sum_i \exp\left(-\frac{\tilde{T}_i^{(r)} \varepsilon}{2nL}\right) \leq \delta/4. \quad (58)$$

□

We next bound the error in our linear approximation, the first order error term in (56), showing that when our termination condition is met the error in our linear term is at most  $\varepsilon/2$ .

**Lemma 8.** *When sampling according to  $\tilde{T}_i^{(r)}$  which satisfy*

$$\sum_{i=1}^n \frac{g_i^2 \sigma_i^2}{\tilde{T}_i^{(r)}} + \frac{\varepsilon}{3} \max_i \frac{g_i}{\tilde{T}_i^{(r)}} \leq \frac{\varepsilon^2}{8 \log(8/\delta)},$$

we have that

$$\mathbb{P}\left(|\nabla f(\boldsymbol{\mu})^\top (\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu})| \geq \varepsilon/2\right) \leq \delta/4.$$

*Proof.* To prove this lemma we utilize Bernstein's inequality, which states that if  $X_1, \dots, X_n$  are independent zero-mean random variables, where  $|X_i| \leq M$  almost surely for all  $i$ , then for all  $\varepsilon > 0$

$$\mathbb{P}\left(\left|\sum_{i=1}^n X_i\right| \geq t\right) \leq 2 \exp\left(-\frac{\varepsilon^2/2}{\sum_{i=1}^n \mathbb{E}\{X_i^2\} + M\varepsilon/3}\right). \quad (59)$$

Denoting the  $j$ -th pull of arm  $i$  as  $Z_{i,j} \stackrel{\text{i.i.d.}}{\sim} Z_i$ , where  $Z_i$  is the distribution of arm  $i$ , we have for all  $i, j$  that  $\mathbb{E}\{Z_{i,j}\} = \mu_i$ ,  $\text{Var}(Z_{i,j}) = \sigma_i^2$ , and that  $|Z_{i,j}| \leq 1$  almost surely. Since  $\mathbb{E}\{\nabla f(\boldsymbol{\mu})^\top (\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu})\} = 0$  we can apply Bernstein's inequality on these  $\sum_{i=1}^n \tilde{T}_i^{(r)}$  random variables, yielding

$$\begin{aligned} \mathbb{P}\left(|\nabla f(\boldsymbol{\mu})^\top (\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu})| \geq \varepsilon/2\right) &= \mathbb{P}\left(\left|\sum_{i=1}^n \sum_{j=1}^{\tilde{T}_i^{(r)}} \frac{g_i}{\tilde{T}_i^{(r)}} (Z_{i,j} - \mu_i)\right| \geq \varepsilon/2\right) \\ &\leq 2 \exp\left(-\frac{\varepsilon^2/8}{\sum_{i=1}^n \frac{g_i^2 \sigma_i^2}{\tilde{T}_i^{(r)}} + \frac{\varepsilon}{3} \max_i \frac{g_i}{\tilde{T}_i^{(r)}}}\right). \end{aligned} \quad (60)$$

As claimed, this is less than  $\delta/4$  when

$$\sum_{i=1}^n \frac{g_i^2 \sigma_i^2}{\tilde{T}_i^{(r)}} + \frac{\varepsilon}{3} \max_i \frac{g_i}{\tilde{T}_i^{(r)}} \leq \frac{\varepsilon^2}{8 \log(8/\delta)}. \quad (61)$$

□

Combining together Lemmas 7 and 8, we have the following lemma.

**Lemma 9.** *If*

$$\sum_{i=1}^n \frac{g_i^2 \sigma_i^2}{\tilde{T}_i^{(r)}} + \frac{\varepsilon}{3} \max_i \frac{g_i}{\tilde{T}_i^{(r)}} \leq \frac{\varepsilon^2}{8 \log(8/\delta)}$$

then with probability at least  $1 - \delta/2$

$$|f(\tilde{\boldsymbol{\mu}}) - f(\boldsymbol{\mu})| \leq \varepsilon.$$

## D.2 Bounding the sample complexity

The drawback of Lemma 9 is that it requires knowledge of the true gradient  $\nabla f(\boldsymbol{\mu})$  and the noise variances. Since we know neither  $\{\sigma_i\}$  nor  $\{g_i\}$ , we must instead construct a termination condition in Algorithm 3 using our empirical estimates of these quantities. We do this by considering the worst feasible instance within our confidence intervals, as in Algorithm 1, since we then have that conditioned on  $\xi$

$$\sum_{i=1}^n \frac{g_i^2 \sigma_i^2}{\tilde{T}_i^{(r)}} + \frac{\varepsilon}{3} \max_i \frac{g_i}{\tilde{T}_i^{(r)}} \leq \sum_{i=1}^n \frac{\left(\hat{g}_i^{(r,U)} \hat{\sigma}_i^{(r,U)}\right)^2}{\tilde{T}_i^{(r)}} + \frac{\varepsilon}{3} \max_i \frac{\hat{g}_i^{(r,U)}}{\tilde{T}_i^{(r)}}, \quad (62)$$

where this right hand side is our computable stopping condition in Algorithm 3 which we compare with  $\frac{\varepsilon^2}{8 \log(8/\delta)}$ .

We bound the time it takes for this termination condition to be reached in the following lemma. We begin by analyzing the number of samples required for the second term in (62) to be sufficiently small.

**Lemma 10.** *On  $\xi$ , Algorithm 3 must terminate and produce  $\tilde{T}_i^{(r)}$  such that*

$$\frac{\varepsilon}{3} \max_i \frac{\hat{g}_i^{(r,U)}}{\tilde{T}_i^{(r)}} \leq \frac{\varepsilon^2}{16 \log(8/\delta)}$$

when

$$B_r \geq \frac{13nL^{2/3} \log(32nr^2/\delta)}{\varepsilon^{2/3}} + \frac{48n \max_i g_i \log(8/\delta)}{\varepsilon}.$$

*Proof.* We use the same techniques as in the proof of Theorem 1, bounding our expression as

$$\begin{aligned} \frac{\varepsilon}{3} \max_i \frac{\hat{g}_i^{(r,U)}}{\tilde{T}_i^{(r)}} &\stackrel{(a)}{\leq} \frac{\varepsilon}{3} \max_i \frac{g_i + 2LC_\mu(i, r)}{\tilde{T}_i^{(r)}} \\ &\leq \frac{\varepsilon}{3} \cdot \max_i \frac{g_i + 2L \left( \sqrt{\frac{2 \log(32nr^2/\delta)}{B_r/n}} + \frac{7 \log(32nr^2/\delta)}{3(B_r/n-1)} \right)}{B_r/n} \\ &\leq \frac{\varepsilon}{3} \left( \frac{\max_i g_i + 8L \sqrt{\frac{\log(32nr^2/\delta)}{B_r/n}}}{B_r/n} \right), \end{aligned} \quad (63)$$

where (a) utilizes the event  $\xi$ . We see that this expression is less than  $\frac{\varepsilon^2}{16 \log(8/\delta)}$  when

$$B_r \geq \frac{13nL^{2/3} \log(32nr^2/\delta)}{\varepsilon^{2/3}} + \frac{48n \max_i g_i \log(8/\delta)}{\varepsilon}. \quad (64)$$

□

We now examine the number of samples required for the dominant portion of our stopping condition, the first term in (62), to be sufficiently small. For notational simplicity we define our new measure of problem complexity  $H \triangleq \sum_i g_i \sigma_i$ .

**Lemma 11.** *On  $\xi$ , Algorithm 3 will produce  $\tilde{T}_i^{(r)}$  such that*

$$\sum_{i=1}^n \frac{\left(\hat{g}_i^{(r,U)} \hat{\sigma}_i^{(r,U)}\right)^2}{\tilde{T}_i^{(r)}} \leq \frac{\varepsilon^2}{16 \log(8/\delta)}$$

when

$$B_r = \Omega \left( \frac{H^2 \log(1/\delta)}{\varepsilon^2} + \left[ \frac{n\sqrt{\sum_i g_i^2} + nL\sqrt{\sum_i \sigma_i^4}}{\varepsilon} + \frac{(nL \sum_i \sigma_i)^{2/3}}{\varepsilon^{2/3}} + \frac{\sqrt{Ln^{5/4}}}{\sqrt{\varepsilon}} \right] \log(nr^2/\delta) \right).$$

*Proof.* Examining when this termination is met, we need to upper bound terms in the right hand side of (62). We do so by analyzing four separate cases, based on the magnitudes of  $g_i$  and  $\sigma_i$ . Our primary term of consideration is when both are relatively large, in which case  $\alpha_i$  (dropping the implicit  $r$  superscript) are order optimal sampling frequencies. For ease of notation, we define the set of coordinates with large  $g_i$  as  $\mathcal{S}_g \triangleq \{i : g_i \geq 4LC_\mu(i, r)\}$ , and the corresponding set  $\mathcal{S}_\sigma \triangleq \{i : \sigma_i \geq 2C_\sigma(i, r)\}$ , where we note that for those indices  $\mathcal{S} \triangleq \{i : g_i \geq 4LC_\mu(i, r) \text{ and } \sigma_i \geq 2C_\sigma(i, r)\} = \mathcal{S}_g \cap \mathcal{S}_\sigma$ , we have  $\tilde{T}_i^{(r)} \geq \alpha_i B_r \geq \frac{g_i \sigma_i}{4H} B_r$  for  $i \in \mathcal{S}$ . We note that due to our second round of sampling, we have that  $T_i^{(r)} \geq \beta_i^{(r)} B_r \geq \frac{\sigma_i}{2\sum_j \sigma_j} B_r$  for  $i \in \mathcal{S}_\sigma$ , and that for all  $i$  we have  $T_i^{(r)} \geq B_r/n$  and  $\tilde{T}_i^{(r)} \geq B_r/n$ . We begin by splitting our summation into the 4 terms discussed.

$$\begin{aligned} & \sum_{i=1}^n \frac{\left(\hat{g}_i^{(r,U)} \hat{\sigma}_i^{(r,U)}\right)^2}{\tilde{T}_i^{(r)}} \\ & \leq \sum_{i=1}^n \frac{(g_i + 2LC_\mu(i, r))^2 (\sigma_i + C_\sigma(i, r))^2}{\tilde{T}_i^{(r)}} \\ & \leq \sum_{i \in \mathcal{S}} \frac{6g_i^2 \sigma_i^2}{\tilde{T}_i^{(r)}} + \sum_{\mathcal{S}_g \setminus \mathcal{S}} \frac{7g_i^2 C_\sigma^2(i, r)}{\tilde{T}_i^{(r)}} + \sum_{\mathcal{S}_\sigma \setminus \mathcal{S}} \frac{81L^2 C_\mu^2(i, r) \sigma_i^2}{\tilde{T}_i^{(r)}} + \sum_{i \in \mathcal{S}_\sigma \cup \mathcal{S}_g} \frac{324L^2 C_\mu^2(i, r) C_\sigma^2(i, r)}{\tilde{T}_i^{(r)}} \end{aligned} \quad (65)$$

We now bound each of these terms sequentially. We begin by noting that on  $\xi$

$$\sum_{i \in \mathcal{S}} \frac{g_i^2 \sigma_i^2}{\tilde{T}_i^{(r)}} \leq \frac{H^2}{4B_r}, \quad (66)$$

which is our primary error term.

Our second term stems from when  $g_i$  is large but  $\sigma_i$  is small, where we utilize the uniform sampling based lower bound on the number of pulls. Here we see that

$$\sum_{\mathcal{S}_g \setminus \mathcal{S}} \frac{g_i^2 C_\sigma^2(i, r)}{\tilde{T}_i^{(r)}} = \sum_{\mathcal{S}_g \setminus \mathcal{S}} \frac{2g_i^2 \log(8nr^2/\delta)}{\tilde{T}_i^{(r)} \cdot (T_i^{(r)} - 1)} = O \left( \frac{n^2 (\sum_i g_i^2) \log(nr^2/\delta)}{B_r^2} \right). \quad (67)$$

Our third term occurs when  $\sigma_i$  is large but  $g_i$  is small, in which case we utilize the fact that  $T_i^{(r)} \geq \frac{\sigma_i}{2\sum_j \sigma_j} B_r$  sample proportionally to  $\sigma_i$ . Using the fact that on  $\xi$  our estimators stay within their confidence intervals, we can bound this sum as

$$\begin{aligned} \sum_{\mathcal{S}_\sigma \setminus \mathcal{S}} \frac{L^2 C_\mu^2(i, r) \sigma_i^2}{\tilde{T}_i^{(r)}} & \leq \sum_{\mathcal{S}_\sigma \setminus \mathcal{S}} \frac{L^2 \left( \sigma_i \sqrt{\frac{2 \log(32nr^2/\delta)}{T_i^{(r)}} + \frac{13 \log(32nr^2/\delta)}{3(T_i^{(r)} - 1)}} \right)^2 \sigma_i^2}{\tilde{T}_i^{(r)}} \\ & = O \left( \sum_{\mathcal{S}_\sigma \setminus \mathcal{S}} \frac{L^2 \left( \frac{\sigma_i^2 \log(nr^2/\delta)}{T_i^{(r)}} + \frac{\log^2(nr^2/\delta)}{(T_i^{(r)})^2} \right) \sigma_i^2}{\tilde{T}_i^{(r)}} \right) \\ & \leq O \left( L^2 \sum_{\mathcal{S}_\sigma \setminus \mathcal{S}} \left( \frac{\sigma_i^4 \log(nr^2/\delta)}{T_i^{(r)} \cdot \tilde{T}_i^{(r)}} + \frac{\sigma_i^2 \log^2(nr^2/\delta)}{(T_i^{(r)})^2 \cdot \tilde{T}_i^{(r)}} \right) \right) \\ & \leq O \left( \frac{n^2 L^2 (\sum_i \sigma_i^4) \log(nr^2/\delta)}{B_r^2} + \frac{n^2 L^2 (\sum_i \sigma_i)^2 \log^2(nr^2/\delta)}{B_r^3} \right). \end{aligned} \quad (68)$$

Our final term is when both  $\sigma_i$  and  $g_i$  are small. In this case  $\sigma_i^2 \leq 2C_\sigma(i, r) = \frac{2 \log(8nr^2/\delta)}{T_i^{(r)} - 1}$ , and so we can simplify the sum as

$$\begin{aligned} \sum_{i \in \bar{\mathcal{S}}_\sigma \cup \bar{\mathcal{S}}_g} \frac{L^2 C_\mu^2(i, r) C_\sigma^2(i, r)}{\tilde{T}_i^{(r)}} &\leq \sum_{i \in \bar{\mathcal{S}}_\sigma \cup \bar{\mathcal{S}}_g} \frac{L^2 \left( \sigma_i \sqrt{\frac{2 \log(32nr^2/\delta)}{T_i^{(r)}} + \frac{13 \log(32nr^2/\delta)}{3(T_i^{(r)} - 1)}} \right)^2 \frac{2 \log(8nr^2/\delta)}{T_i^{(r)} - 1}}{\tilde{T}_i^{(r)}} \\ &= O \left( \sum_{i \in \bar{\mathcal{S}}_\sigma \cup \bar{\mathcal{S}}_g} \frac{L^2 \log^3(nr^2/\delta)}{\tilde{T}_i^{(r)} (T_i^{(r)})^3} \right) \\ &\leq O \left( \frac{L^2 n^5 \log^3(nr^2/\delta)}{B_r^4} \right). \end{aligned} \quad (69)$$

We now see that the left hand side of our termination condition is upper bounded by plugging (66), (67), (68), and (69) into (65), yielding

$$\begin{aligned} \sum_{i=1}^n \frac{(\hat{g}_i^{(r,U)} \hat{\sigma}_i^{(r,U)})^2}{\tilde{T}_i^{(r)}} &= \\ O \left( \frac{H^2}{B_r} + \frac{n^2 (\sum_i g_i^2) \log(nr^2/\delta)}{B_r^2} + \frac{n^2 L^2 (\sum_i \sigma_i^4) \log(nr^2/\delta)}{B_r^2} + \frac{n^2 L^2 (\sum_i \sigma_i)^2 \log^2(nr^2/\delta)}{B_r^3} + \frac{L^2 n^5 \log^3(nr^2/\delta)}{B_r^4} \right). \end{aligned} \quad (70)$$

We can terminate as soon as this quantity is less than  $\frac{\varepsilon^2}{16 \log(8/\delta)}$ , which must happen when

$$B_r = \Omega \left( \frac{H^2 \log(1/\delta)}{\varepsilon^2} + \left[ \frac{n \sqrt{(\sum_i g_i^2)}}{\varepsilon} + \frac{nL \sqrt{(\sum_i \sigma_i^4)}}{\varepsilon} + \frac{(nL \sum_i \sigma_i)^{2/3}}{\varepsilon^{2/3}} + \frac{\sqrt{Ln^{5/4}}}{\sqrt{\varepsilon}} \right] \log(nr^2/\delta) \right). \quad (71)$$

The final thing to note is that our algorithm starts with a budget of  $B_0$ , and so the total number of samples used must be at least  $B_0$ .  $\square$

With these lemmas in hand, the proof of Theorem 3 is straightforward.

*Proof of Theorem 3.* The error of the plug-in estimator  $f(\tilde{\boldsymbol{\mu}})$  satisfies

$$\begin{aligned} \mathbb{P}(|f(\tilde{\boldsymbol{\mu}}) - f(\boldsymbol{\mu})| > \varepsilon) &= \mathbb{P}(\bar{\xi}) + \mathbb{P}(|f(\tilde{\boldsymbol{\mu}}) - f(\boldsymbol{\mu})| > \varepsilon \mid \xi) \\ &\leq \mathbb{P}(\bar{\xi}) + \mathbb{P}(|\nabla f(\boldsymbol{\mu})^\top (\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu})| > \varepsilon/2 \mid \xi) + \mathbb{P}\left(\frac{L}{2} \|\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_2^2 > \varepsilon/2 \mid \xi\right) \\ &\leq \delta, \end{aligned} \quad (72)$$

by utilizing Lemmas 6 and 9, as on  $\xi$  Algorithm 3 will not terminate until the conditions of Lemma 9 are met.

Analyzing the number of arm pulls required for this algorithm to succeed, we have by Lemmas 10 and 11 that the algorithm will terminate when

$$B_r = \Omega \left( \frac{H^2 \log(1/\delta)}{\varepsilon^2} + \left[ \frac{n \sqrt{(\sum_i g_i^2)}}{\varepsilon} + \frac{nL \sqrt{(\sum_i \sigma_i^4)}}{\varepsilon} + \frac{(nL \sum_i \sigma_i)^{2/3} + nL^{2/3}}{\varepsilon^{2/3}} + \frac{\sqrt{Ln^{5/4}}}{\sqrt{\varepsilon}} \right] \log(nr^2/\delta) \right). \quad (73)$$

Since our algorithm makes at most  $3B_\ell + 2n$  pulls during round  $\ell$ , and at most  $4B_r + n$  pulls to construct  $\tilde{\boldsymbol{\mu}}$  if the termination condition is met in round  $r$ , it in total makes no more than  $10B_r$  pulls before terminating at the end of round  $r$ .



Converting this to an overall budget, we see that the  $\log(r)$  term generates additional  $\log \log$  factors due to the doubling round budgets, as in Theorem 2. This yields an overall budget of

$$\begin{aligned} & O\left(\frac{H^2 \log(1/\delta)}{\varepsilon^2} + \frac{n^2 L \log(n/\delta)}{\varepsilon}\right) \\ & + \tilde{O}\left(\left[\frac{n\sqrt{(\sum_i g_i^2)}}{\varepsilon} + \frac{nL\sqrt{(\sum_i \sigma_i^4)}}{\varepsilon} + \frac{(nL \sum_i \sigma_i)^{2/3} + nL^{2/3}}{\varepsilon^{2/3}} + \frac{\sqrt{Ln}^{5/4}}{\sqrt{\varepsilon}}\right] \log(n/\delta)\right), \end{aligned} \quad (74)$$

where  $\tilde{O}$  suppresses  $\log \log(\cdot)$  terms in  $\text{poly}(n, \delta, \sum_i g_i, \sum_i \sigma_i, \varepsilon^{-1})$ .  $\square$

### D.3 Improved $\alpha_i^{(r)}$

As previously mentioned, the frequencies  $\alpha$  in Algorithm 3 are designed with simplicity and asymptotic optimality in mind. By taking

$$\alpha_i^{(r)} = \frac{\hat{g}_i^{(r,L)} \hat{\sigma}_i^{(r,L)}}{\sum_j \hat{g}_j^{(r,L)} \hat{\sigma}_j^{(r,L)}} + \frac{\hat{g}_i^{(r,L)}}{\sum_j \hat{g}_j^{(r,L)}} + \frac{\left(\hat{\sigma}_i^{(r,L)}\right)^2}{\sum_j \left(\hat{\sigma}_j^{(r,L)}\right)^2} + \frac{1}{n}, \quad (75)$$

we are able to give tighter bounds on our lower order terms, eliminating dependencies on  $\sum_i \sigma_i^4$  and  $\sum_i g_i^2$ .

Concretely, the sample complexity bound can be improved to

$$\begin{aligned} & O\left(\frac{H^2 \log(1/\delta)}{\varepsilon^2} + \frac{n^2 L \log(n/\delta)}{\varepsilon}\right) \\ & + \tilde{O}\left(\left[\frac{\sqrt{n}(\sum_i g_i) + \sqrt{n}L(\sum_i \sigma_i^2)}{\varepsilon} + \frac{(nL \sum_i \sigma_i)^{2/3} + nL^{2/3}}{\varepsilon^{2/3}} + \frac{\sqrt{Ln}^{5/4}}{\sqrt{\varepsilon}}\right] \log(n/\delta)\right). \end{aligned} \quad (76)$$

This is achieved by using that  $\tilde{T}_i^{(r)} \geq \frac{g_i}{2 \sum_j g_j}$  for large enough  $g_i$  in (67), and that  $\tilde{T}_i^{(r)} \geq \frac{\sigma_i^2}{2 \sum_j \sigma_j^2}$  for large enough  $\sigma_i$  in (68), as sampling proportionally to  $\sigma_i^2$  minimizes the maximal confidence interval width.

This only increases the sample complexity by at most a factor of 3 as we are doubling the amount of sampling we perform in the last iteration, and performing a ceiling operation.

### D.4 Gain of variance adaptivity

In the case with unknown variance and gradient information, we see that our optimal sampling frequencies are  $\alpha_i \propto g_i \sigma_i$ , yielding a first order error of

$$\mathcal{N}\left(0, \frac{1}{T} \sum_{i=1}^n \frac{g_i^2 \sigma_i^2}{\alpha_i}\right) \sim \mathcal{N}\left(0, \frac{1}{T} \left(\sum_{i=1}^n g_i \sigma_i\right)^2\right). \quad (77)$$

Hence, we have that

$$O\left(\left(\sum_i g_i \sigma_i\right)^2 \varepsilon^{-2} \log(1/\delta)\right) + o(\varepsilon^{-2}) \quad (78)$$

samples are sufficient for an adaptive algorithm to achieve  $\varepsilon$  error. For a nonadaptive scheme, we see by a similar argument that the first order error of such a scheme is distributed as

$$\mathcal{N}\left(0, \frac{n}{T} \sum_{i=1}^n g_i^2 \sigma_i^2\right), \quad (79)$$

requiring

$$O\left(n \left(\sum_i g_i^2 \sigma_i^2\right) \varepsilon^{-2} \log(1/\delta)\right) + o(\varepsilon^{-2}) \quad (80)$$

samples to achieve  $\varepsilon$  error. Hence, the gain afforded by adaptivity is

$$O\left(\frac{n \sum_i g_i^2 \sigma_i^2}{(\sum_i g_i \sigma_i)^2}\right). \quad (81)$$

## E Lower bound

*Proof of Proposition 1.* For a plug-in estimator  $\hat{\boldsymbol{\mu}}$  generated by sampling the  $i$ -th arm  $T_i$  times, we see that the error  $f(\boldsymbol{\mu}) - f(\hat{\boldsymbol{\mu}})$  is distributed as

$$\mathbf{g}^\top (\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}) \sim \mathcal{N}\left(0, \sum_i \frac{g_i^2 \sigma_i^2}{T_i}\right) \quad (82)$$

when arm pulls from arm  $i$  are corrupted by independent additive Gaussian noise with variance  $\sigma_i^2$ . We then have by standard Gaussian anti-concentration inequalities that  $\mathbb{P}(\mathcal{N}(0, 1) > x) \geq \frac{1}{4}e^{-x^2}$ , and so

$$\mathbb{P}\left(\mathcal{N}\left(0, \sum_i \frac{g_i^2 \sigma_i^2}{T_i}\right) \geq \varepsilon\right) \geq \frac{1}{4} \exp\left(-\frac{\varepsilon^2}{\sum_i \frac{g_i^2 \sigma_i^2}{T_i}}\right), \quad (83)$$

where in order for this to be less than  $\delta$ , it must be the case that

$$\mathbb{P}\left(\mathcal{N}\left(0, \sum_i \frac{g_i^2 \sigma_i^2}{T_i}\right) \geq \varepsilon\right) < \delta \implies \sum_i \frac{g_i^2 \sigma_i^2}{T_i} \leq \frac{\varepsilon^2}{\log(1/4\delta)}. \quad (84)$$

Rewriting this as  $T_i = \alpha_i T$ , where  $\boldsymbol{\alpha}$  is the sampling probability vector, we have as before that

$$\min_{\boldsymbol{\alpha} \in \Delta^n} \sum_i \frac{g_i^2 \sigma_i^2}{\alpha_i T} = \frac{1}{T} \left(\sum_i g_i \sigma_i\right)^2 \quad (85)$$

by taking the Lagrangian, invoking Sion's minimax theorem [Sion, 1958], and optimizing to yield  $\alpha_i \propto g_i \sigma_i$ . Plugging this back in, we see that

$$\sum_i \frac{g_i^2 \sigma_i^2}{\alpha_i T} \leq \frac{\varepsilon^2}{\log(1/4\delta)} \implies T \geq \left(\sum_i g_i \sigma_i\right)^2 \varepsilon^{-2} \log(1/4\delta). \quad (86)$$

□

## F Experimental Details

We implemented all algorithms in Python. The results and figures in this paper can be reproduced from the available code. We now describe important practical optimizations for our adaptive methods to outperform the optimism-based fully sequential method and batched uniform scheme. All experiments were run on one core of an AMD Opteron Processor 6378 with 500GB memory (no parallelism within a trial). No empirical standard deviations were included, as all simulations were run for 100 trials for the optimism based fully sequential method and 1000 trials for the batched methods. Due to the low variance in our outputs, the confidence intervals around our average error are extremely small. We use  $\delta = .01$  for all experiments.

### F.1 Additional numerical results

For generating data for our  $Ax$  experiments, we construct our ground truth  $\boldsymbol{\mu}$  vector as drawn from a pareto(.5) distribution, then normalize it so the largest entry is 20, to maintain a reasonable signal to noise ratio. We then construct  $A = \boldsymbol{\mu}x^\top / \|x\|_2^2 + Z$ , where the entries of  $x$  are i.i.d. Uniform([.5,1]), and  $Z \in \mathbb{R}^{n \times d}$  has i.i.d. normal entries with variance  $1/d^2$ , to prevent the variances from scaling with the dimension. We correct the final product by updating  $A \leftarrow A - (Ax - \boldsymbol{\mu})\mathbf{1}^\top / \|x\|_1$  to ensure that  $Ax = \boldsymbol{\mu}$ . This functions as desired, as since  $x > 0$  entrywise we have that  $\|x\|_1 = \mathbf{1}^\top x$ . We provide several additional simulations below in Figure 4, corroborating our numerical results in the main body. Note that when there is an underlying computational problem, as in the case of estimating  $\|Ax\|_2^2$ , this underlying computational problem allows for exact computation of certain coordinates, resulting in much larger gains of adaptivity for certain computational budgets.

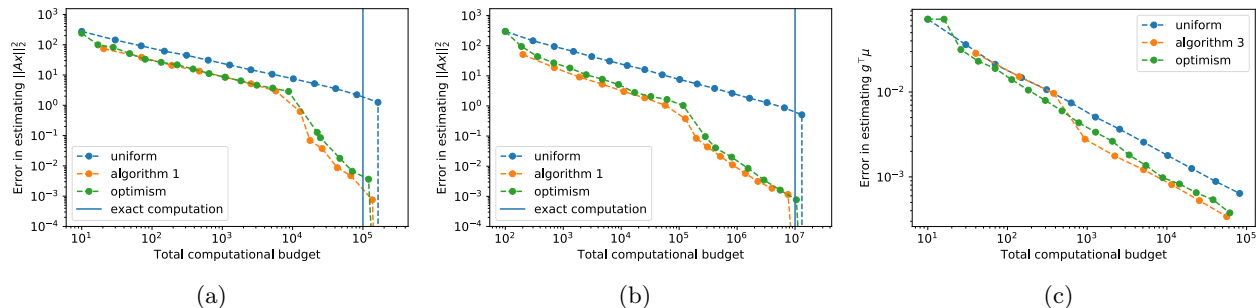


Figure 4: In (a) we have the estimation error of  $\|Ax\|_2^2$  for matrix  $A$ , vector  $x$ , for  $n = 10$  and  $d = 10k$ . The problem instance shown in Figure 4a is projected to have a gain of adaptivity of 7.5. In (b) we see the estimation error of  $\|Ax\|_2^2$  for matrix  $A$ , vector  $x$ , for  $n = 100$  and  $d = 10k$ . In (c) we plot the estimation error of  $\mathbf{g}^\top \boldsymbol{\mu}$  for  $n = 10$ ,  $w$  uniform in  $[0,1]$  and  $\boldsymbol{\mu}$  normalized Pareto distribution.

## F.2 Modifications from written algorithm

In order to obtain good finite sample performance, we make several changes to the algorithms as stated. Firstly, we reuse all samples throughout the algorithm, including in the final outputted  $\tilde{\boldsymbol{\mu}}$ . Secondly, we start from a budget of 1 pull per arm with  $B_0 = n$ , and double from there. Additionally, we use as our confidence interval width  $C_r = \sqrt{\log(1/\delta)/\tilde{B}_r}$ . Finally, we use  $\tilde{T}_i^{(r)} = \lceil \min(r, 10)\alpha_i B_r \rceil$ , where uniform sampling is not practically necessary due to the reuse of samples, and the factor of 10 allows our algorithm to perform better in terms of constants. Our theoretical analysis can accommodate this factor of 10. Note that this factor of 10 directly translates into the slope of the linear relationship; since the algorithm will have made approximately  $2B_r$  uniform samples by the end of round  $r$ , it is only able to allocate  $10/(10+2) \approx .83$  fraction of its samples to the optimal sampling frequencies. This is empirically validated by our simulations, as we can see that the slope of the line of best fit is indeed approximately .83. This parameter of 10 needs to be tuned carefully in practice, as it indicates how greedy the algorithm will be; if this factor is chosen to be too large, then the algorithm will ignore higher order effects and act too strongly on the imperfect gradient information. For the plots showing the error as a function of number of samples, we take all the uniform samples up to round  $r$  for the adaptive algorithm, then generate the termination sampling pattern  $\tilde{T}_i^{(r)}$ , sample according to that, and measure and record the error. These samples, corresponding to the  $\tilde{T}_i^{(r)}$ , are then discarded, and uniform samples for round  $r+1$  are taken.

For the gain in terms of stopping time of adaptive over uniform sampling, we use slightly perturbed realizations to better display the trend, and avoid edge / rounding effects. To this end, we modify our algorithms to utilize the round budget  $B_r = n(2-c)^r$ . We then iterate over this base, ranging  $c$  from  $c_0 = 2$  to  $c_{99} = \frac{301}{200}$  with  $c_i = 2 - \frac{i}{200}$  for 100 trials. No failures were recorded in the stopping time simulations; that is, all outputted  $\tilde{\boldsymbol{\mu}}$  satisfied  $|f(\tilde{\boldsymbol{\mu}}) - f(\boldsymbol{\mu})| \leq \varepsilon$  for both the batched uniform and our adaptive algorithms. Additionally, the stopping conditions yielded comparable error for both methods, with average outputted error within a factor of 2 for the uniform and adaptive methods across all settings tested.

We implement the optimism-based algorithm of [Carpentier et al., 2015] with a priority queue to ensure efficient selection of an arm at time  $t$ , requiring amortized  $\log n$  time instead of  $O(n)$  time. We additionally generate all possible Bernoulli samples ahead of time, and within our timed experiments only query the pre-drawn samples. Confidence intervals are used with  $\beta = \sqrt{\log(1/\delta)}$ .

In order to make the algorithms efficient in the  $\|Ax\|_2^2$  example, we first randomly permute the columns of  $A$  and entries of  $x$  jointly, and then take our arm pulls as iterating over these entries in order. This allows the uniform arm pulls in our batched algorithms to be taken as BLAS efficient operations, simply taking the matrix vector product between a column subsampled  $A$  and  $x$ . Note that this can have statistical benefits from correlating the noise in our sampling, as discussed in [Baharav and Tse, 2019].

### F.3 Optimism based algorithm

We describe our optimism based algorithm below for estimating  $f(\boldsymbol{\mu}) = \|\boldsymbol{\mu}\|_2^2$  in the case of  $\|Ax\|_2^2$ . When we refer to pulling arm  $i$ , we mean selecting a random coordinate  $J \sim \text{Unif}([d])$  and computing  $A_{i,J}x_J$ . We utilize a priority queue on the  $B_{i,t}$ , noting that only  $B_{i_t^*,t}$  will change at any given time step. This means that every iteration of this algorithm can be performed in amortized  $O(\log n)$  time, instead of the naive  $O(n)$  complexity of recomputing the  $B_{i,t}$  from scratch in every iteration and iterating over to find the largest.

---

#### Algorithm 4 Optimism based algorithm for $\|Ax\|_2^2$

---

- 1: **Input:** arms  $[n]$ , matrix  $A \in \mathbb{R}^{n \times d}$ , vector  $x \in \mathbb{R}^d$ , target accuracy  $\varepsilon$ , error probability  $\delta$
  - 2: Pull each arm once
  - 3: **for**  $t = n + 1, \dots, T$  **do**
  - 4:     Compute for each arm  $i \in [n]$ 

$$B_{i,t} = \frac{1}{T_{i,t}} \left( \hat{\mu}_{i,t} + \frac{\sqrt{\log(1/\delta)}}{T_{i,t}} \right)$$
  - 5:     Select  $i_t^* \in \operatorname{argmax}_{i: T_{i,t} \leq d} B_{i,t}$
  - 6:     **if**  $T_{i_t^*,t} = d$  **then**
  - 7:         Exactly compute  $\mu_{i_t^*} = A_{i_t^*}^\top x$
  - 8:     **else**
  - 9:         Pull arm  $i_t^*$
  - 10:     **end if**
  - 11:     **if**  $T_{i,t} = d$  for all  $i$  **then**
  - 12:         **return** Exactly computed  $\|\hat{\boldsymbol{\mu}}_t\|_2^2 = \|Ax\|_2^2$
  - 13:     **end if**
  - 14: **end for**
  - 15: **return**  $\sum_{i=1}^n \hat{\mu}_{i,t}^2 - \frac{\mathbb{1}_{\{T_{i,t} < d\}}}{T_{i,t}}$
-