Batched Stochastic Linear Bandits with 1-Bit Communication Constraints

Ivan Lau
National University of Singapore
ivan.lau@u.nus.edu

Daniel McMorrow
National University of Singapore
dmcmorrow@u.nus.edu

Kevin Jamieson
University of Washington
jamieson@cs.washington.edu

Jonathan Scarlett
National University of Singapore
dcsjms@nus.edu.sg

Abstract

We study stochastic linear bandits under a natural combination of batching and communication constraints: the time horizon is partitioned into batches of equal size B, and during each batch the learner sends B requested arm pulls to an agent, who then observes the corresponding B rewards and responds with a single bit of feedback to the learner. For each batch, the learner specifies the 1-bit quantization rule the agent uses, which may depend on all previously received bits but not on any past rewards directly. This setting addresses a significant yet unexplored "middle ground" between previous models having per-round quantization only *or* total bit budgets only. We establish a minimax lower bound showing that in d dimensions with action set size $|\mathcal{A}|$, $\Omega(\min\{T, B\min\{d, \log|\mathcal{A}|\}\})$ regret is unavoidable due to the 1-bit communication bottleneck, even in the absence of noise. Combined with standard statistical limits, this yields a general lower bound of $\widetilde{\Omega}(\min\{T,\ B\min\{d, \log|\mathcal{A}|\} + \sqrt{dT\min\{d, \log|\mathcal{A}|\}}\})$. We develop two phased-elimination algorithms based on G-optimal designs and 1-bit mean estimation. The first achieves $\widetilde{O}(dB + d\sqrt{T})$ regret, matching the lower bound up to logarithmic factors when $|\mathcal{A}| = \exp(\Omega(d))$, and the second incorporates a safe-arm identification and warm-start procedure to obtain $\widetilde{O}(B\log|\mathcal{A}| + d^{3/2}\sqrt{B} + \sqrt{dT\log|\mathcal{A}|})$ regret, which is near-optimal in broad scaling regimes of $(|\mathcal{A}|, B, d, T)$. Together, our results demonstrate that a single bit of feedback per batch suffices to nearly match the minimax regret of *unconstrained* linear bandits in broad scaling regimes, even for batch sizes as large as $\Theta(\sqrt{T})$.

1 Introduction

The study of bandit algorithms has seen a resurgence over the past two decades, driven by applications in online advertising (Li et al., 2010; Mehta et al., 2007), clinical trials (Durand et al., 2018), anomaly detection (Ding et al., 2019; Soemers et al., 2018), and recommendation systems (Mary et al., 2015; Bouneffouf et al., 2012). Within this framework, we consider a sequential decision-making problem arising from a fundamental asymmetry in communication bandwidth. A *learner* coordinates with an *agent* in the field: the learner can transmit detailed instructions prescribing a batch of B measurements for the agent to take, but the agent—after observing all B outcomes—can respond with only a single bit of feedback. This model is motivated by settings where devices observing rewards have severely limited communication capacity and cannot accumulate memory between interactions. In sensor networks used for environmental monitoring or industrial diagnostics, for instance, each sensor may be battery-constrained and able to transmit only a limited number of bits per reporting interval, and

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

its on-board memory may be too small to store a full history of measurements. Similarly, in federated learning and edge computing deployments, a central server orchestrates experiments by dispatching actions to end devices—phones, IoT endpoints, and similar devices—which execute them over a fixed window before returning a short summary; crucially, the device participating in one round may differ from the device participating in the next, precluding any accumulation of local state. In both cases, single-device-with-unlimited-memory models such as (Salgia and Zhao, 2023; Amani et al., 2023) and per-pull few-bits models such as (Hanna et al., 2022; Mitra et al., 2023; Lau and Scarlett, 2025) (outlined in more detail below) may not fully capture the relevant constraints.

We study the above setting under the standard stochastic linear bandit model, in which each reward is a linear function of an unknown parameter vector corrupted by noise. Our goal is to characterize the fundamental limits of learning under this one-bit uplink bottleneck, and to design algorithms whose performance approaches those limits. Specifically, we focus on *regret minimization*: minimizing the cumulative sub-optimality gap—the difference in mean reward between the best action and the action played—over a fixed time horizon T. In the standard linear bandit setting (Lattimore and Szepesvári, 2020, Chapter 19), the learner directly observes the generated rewards in each time step, and can use this information to guide decision-making in future time steps. Two prominent lines of work studying the role of communication constraints in linear bandits are as follows:

- In works such as (Hanna et al., 2022; Mitra et al., 2023; Lau and Scarlett, 2025), a *memoryless* decentralized agent observes each reward, and based on that reward alone, it sends back a few bits of information (e.g., 3 or 4 bits), i.e., a quantization of the reward.
- In works such as (Salgia and Zhao, 2023; Amani et al., 2023), the agent(s) observing the rewards has *unlimited memory*,[1] and the communication constraint is on the *total number of bits* sent throughout the time horizon. This allows each agent to perform local estimation and send larger amounts of information at suitably-chosen times (while sending nothing at the other times).

As hinted above, the main premise of our work is that these settings leave a *considerable unexplored "middle ground"* in which the agent has *limited memory* and communication occurs at *regular intervals*, but not every round. This setting leads to both distinct algorithmic ideas and new insights, such as just *one bit at the end of each $\sqrt{T}$-round batch* (with time horizon T) often sufficing to match the minimax-optimal regret of unconstrained learning to within logarithmic factors.

Our paper is organized as follows: In Section 1.1, we provide a more detailed summary of existing works related to our problem; in Section 1.2, we outline our main contributions; in Section 2, we give a formal presentation of the problem setup; in Section 3, we formally state our main results; and in Section 4, we detail the corresponding algorithms. The proofs can be found in the appendix.

1.1 Related Work

Linear Bandits. Stochastic linear bandits were first studied in the "standard" setting (i.e., without communication or batch constraints) in (Abe and Long, 1999), and a prominent follow-up work is (Auer, 2002), who considered an algorithm for stochastic linear bandits with a finite action set based on the "optimism principle" (Lattimore and Szepesvári, 2020, Chapter 7.1). (Dani et al., 2008) extended these ideas to provide an algorithm attaining $\widetilde{O}(d\sqrt{T})$ regret for any action set with bounded mean rewards, which they also showed to be minimax optimal up to logarithmic factors for worst-case action sets. The same $\Omega(d\sqrt{T})$ minimax lower bound was subsequently established for the unit sphere and hypercube action sets (Lattimore and Szepesvári, 2020, Chapter 24). Further algorithms for linear bandits with finite action sets were later derived (Lattimore et al., 2020; Chu et al., 2011; Li et al., 2019), all of which (alongside (Auer, 2002)) obtain an upper bound of $\widetilde{O}(\sqrt{dT\log|\mathcal{A}|})$ on the regret, with varying logarithmic dependence. Complementary lower bounds of $\widetilde{\Omega}(\sqrt{dT\min\{d,\log|\mathcal{A}|\}})$ are also well-known for fixed finite action sets (e.g., see (Zhou, 2019; Tajdini et al., 2025)), and $\Omega(\sqrt{dT\log(T/d)\log|\mathcal{A}|})$ for finite action sets that change in each time step (Li et al., 2019).

Bandits with Communication Constraints. Most work on linear bandits assumes that the reward of an action can be observed directly by the learner (with full precision). However, in applications such as sensor networks, rewards are observed by an agent (e.g., sensor) before being communicated to the learner (central server) over a link with limited bandwidth. This has motivated research

[1] They also allow multiple distributed agents, but we are mainly interested in the special case of a single agent.

on communication-constrained bandits; see (Salgia and Zhao, 2023, Appendix A) for a broader survey. The studies most pertinent to this work are those that focus on the quantization of the reward feedback to reduce the uplink communication cost.[2] In particular, (Hanna et al., 2022) devised a quantization scheme that allows existing algorithms to be run with an uplink communication cost of a few bits per arm pull, while maintaining the same regret as the unquantized algorithms up to constant multiplicative factors. An approach based on quantizing the linear parameters (rather than the rewards) was also proposed in (Mitra et al., 2023), attaining $\widetilde{O}(d\sqrt{T})$ regret with an uplink communication cost of $\Theta(d)$ per round. In contrast to these works, our agents have limited memory and can only send one bit after an entire batch of B pulls, which forces the learner to pre-commit actions and reduces the effective communication rate to $1/B$ bits per round. Best-arm identification with one bit per pull has also been studied for quantile rewards (Lau and Scarlett, 2025) and mean rewards (Luong et al., 2026), whereas we study cumulative regret with batched feedback.

Beyond these, communication-constrained bandits have been studied under a variety of related assumptions, including federated linear bandits (Huang et al., 2021; Li and Wang, 2022; He et al., 2022; Fan et al., 2023; Shi et al., 2024) and differentially private linear bandits (Shariff and Sheffet, 2018; Dubey and Pentland, 2020; Hanna et al., 2024) where the privacy constraint induces a communication-like bottleneck. Since the number of such works is large and most operate under assumptions substantially different from ours (e.g., multi-agent coordination, privacy, or unlimited per-agent memory), we focused primarily on the few above that are most directly comparable.

Batched Bandits. Batched bandits were first studied in (Cesa-Bianchi et al., 2013), where they showed that in the multi-armed setting, $O(\log\log T)$ batches suffice to attain order-optimal regret. It was then shown in (Gao et al., 2019) that in order to obtain order-optimal minimax regret in the multi-armed setting, $\Theta(\log\log T)$ batches is both necessary and sufficient for a finite time horizon T. $O(\log\log T)$ batches were also shown to be sufficient to obtain order-optimal regret in the contextual linear bandit setting (Ruan et al., 2021; Zhang et al., 2025; Hanna et al., 2023), of which linear bandits are a special case. Finally, (Ren et al., 2024) proved that 3 batches are necessary and sufficient to attain *asymptotic optimality for a fixed instance* as $T \to \infty$. We also note that, while not explicitly stated as such, the algorithms in (Lattimore et al., 2020; Chu et al., 2011; Li et al., 2019; Salgia and Zhao, 2023) choose multiple arms simultaneously and thus permit varying degrees of batching.

The preceding works allow variable-length batches, whereas *fixed batch lengths* can also have significant practical relevance, and are popular in related topics such as Bayesian optimization (Nava et al., 2022). While fixed batch lengths have been considered in linear bandits, the ones that we are aware of have a different emphasis such as privacy (Chowdhury and Zhou, 2022; Zhou and Chowdhury, 2024), best-arm identification (Jun et al., 2016), or converting *arbitrary* algorithms to batch settings (Provodin et al., 2022). Accordingly, we believe that our findings on regret with fixed batch lengths are of general interest even regardless of communication considerations. We also briefly note that the concept of *delayed feedback* is relevant (Vernade et al., 2020; Howson et al., 2023), since a batch length B implies a delay lying in the range $[0, B]$. Within this setting, we note a result from (Vakili et al., 2023), which when translated to our setup (without 1-bit constraints), yields a regret upper bound of $\widetilde{O}(B + \sqrt{dT\min\{d, \log|\mathcal{A}|\}})$. Compared to the regret in the vanilla linear bandit setup, this introduces an additive "batch or delay penalty" of $\widetilde{O}(B)$. In contrast, Corollary 1 establishes that a significantly larger additive penalty of $\Omega(B\min\{d, \log|\mathcal{A}|\})$ is unavoidable in our setup, showing that communication constraints make our problem strictly harder than standard delayed feedback setting. See also Remark 5 for how increasing the number of feedback bits per batch reduces the communication penalty, while the unavoidable $\Omega(B)$ batching penalty remains.

1.2 Contributions

As noted above, we introduce a new problem setup serving as a "middle ground" between two extremes studied in previous works. Our technical contributions are outlined as follows for an action set $\mathcal{A}$, batch size B, dimension d, and time horizon T (see Section 2 for the detailed problem setup):

- We provide an algorithm that achieves an $|\mathcal{A}|$-independent regret bound that is minimax order-optimal (up to logarithmic factors), even when the noise terms $(\xi_t)_{t=1}^T$ are only assumed to have finite variance (thus potentially being much heavier than sub-Gaussian noise).

[2] We refer to the "uplink communication cost" as the number of bits communicated by the agent to the central learner in a given time step. See Remark 2 for our assumptions on downlink communication.

- We provide a second algorithm that achieves an $|\mathcal{A}|$-dependent regret bound that is order-optimal (up to logarithmic factors) in broad scaling regimes of $(|\mathcal{A}|, B, d, T)$.
- We provide an algorithm-independent minimax lower bound, showing that incurring an expected regret of $\Omega(B \min\{d, \log|\mathcal{A}|\})$ is unavoidable in the worst case. In fact, this bound holds even in the completely noiseless setting ($\sigma = 0$), demonstrating that this bottleneck is purely a consequence of the 1-bit communication constraint. This is combined with well-known lower bounds for standard linear bandits to establish the above near-optimality claims.

Our algorithm for $|\mathcal{A}|$-independent regret is the simpler of the two (though still having numerous technical challenges compared to the standard setting), and involves performing phased elimination with mean rewards that are estimated by *always pulling the same arm B times in a given batch*. In contrast, our algorithm for $|\mathcal{A}|$-dependent regret crucially avoids doing this[3] and instead incorporates initial steps to find a *safe arm* whose regret is "not too high", and partially relies on it to reduce regret.

2 Problem Setup

We consider a stochastic linear bandit instance with dimension $d \in \mathbb{N}$, a finite[4] action set $\mathcal{A} \subset \mathbb{R}^d$, unknown parameter vector $\theta \in \mathbb{R}^d$, and time horizon $T \in \mathbb{N}$. At each time step $t \in \{1, \dots, T\}$, an action $A_t \in \mathcal{A}$ is played, and the reward Y_t is generated according to

$$Y_t = \langle A_t, \theta \rangle + \xi_t, \qquad \mathbb{E}[\xi_t] = 0, \qquad \mathbb{E}[\xi_t^2] \le \sigma^2, \tag{1}$$

where $(\xi_t)_{t=1}^T$ is a sequence of i.i.d. noise terms and $\sigma > 0$ is known. We further assume that mean rewards are *bounded*, in the sense that $|\langle a, \theta \rangle| \le 1$ for each $a \in \mathcal{A}$.

Remark 1 (Effective dimension). For the upper-bound analyses, we may assume full span after expressing the actions and projected parameter in orthonormal coordinates on $\operatorname{span}(\mathcal{A})$. This preserves rewards, regret, cardinality, the communication model, and Euclidean unit-ball constraints. Therefore, Theorems 3 and 4, together with their corollaries, can be improved by replacing d with $\dim \operatorname{span}(\mathcal{A})$, including their dimension-dependent hypotheses. During elimination, we work in the current active span as explained in Remark 6. For lower bounds, Corollary 1 is asserted only for $d \le |\mathcal{A}|$; Theorem 1 has no such restriction.

For a batch size $B \in \{1, \dots, T\}$, we assume the time horizon is split into $M = \lceil T/B \rceil$ equal-sized batches of size B (except possibly a shorter final batch). In each batch $m \in \{1, \dots, M\}$, we have the following interaction between a central learner and an agent:

1. At the beginning of the batch, the learner chooses a length-B sequence of arms $A_{(m-1)B+1}$, $A_{(m-1)B+2}, \dots, A_{mB}$ and a 1-bit quantization function $Q_m : \mathbb{R}^B \to \{0, 1\}$, both of which may depend on the feedback received from batches $1, \dots, m-1$.
2. The agent receives the 1-bit quantization function Q_m, and directly observes the rewards $Y_{(m-1)B+1}, Y_{(m-1)B+2}, \dots, Y_{mB}$, generated according to (1).
3. The agent computes $Q_m(Y_{(m-1)B+1}, Y_{(m-1)B+2}, \dots, Y_{mB})$ and transmits it to the learner.

Remark 2. We do not impose any (downlink) communication constraint from the learner to the agent, motivated by settings where uplink communication poses more of a bottleneck than downlink communication. Although our lower bound allows arbitrary measurable quantizers, our algorithms use only fixed, specified procedures (e.g., comparing empirical averages to specified thresholds, allocating arm pull proportions, and comparing estimated maximum rewards across groups of arms) rather than arbitrary queries. Appendix F summarizes what needs to be sent from the learner to the agent in our algorithms. We also note that while we frame the problem with a single agent for clarity, different batches may correspond to different users or devices. We assume that the agent is 'batch-wise memoryless', meaning its message cannot depend directly on rewards observed in previous batches. Similar assumptions were adopted in previous works (Hanna et al., 2022; Mitra et al., 2023; Mayekar et al., 2023; Lau and Scarlett, 2025; Luong et al., 2026) with $B = 1$ (i.e., no batching). The agent may use temporary memory and computation within a batch, but must follow the action sequence fixed before the batch.

[3] Theorem 2 will reveal that any strategy doing so must be highly suboptimal in regimes of smaller $|\mathcal{A}|$.

[4] Although we formally assume $\mathcal{A}$ is finite to cleanly maintain active sets and compute experimental designs, the regret bounds attained by our $|\mathcal{A}|$-independent algorithm (Theorem 3) also hold for arbitrary compact sets with possibly infinite cardinality via a standard covering argument; see Remark 3 for details.

The goal of the learner is to design an algorithm to minimize the *cumulative regret*, defined as

$$R_T := \sum_{t=1}^{T} \langle a^* - A_t, \theta \rangle, \tag{2}$$

where $a^* := \arg\max_{a \in \mathcal{A}} \langle a, \theta \rangle$ is the optimal arm. We will consider both the expected regret $\mathbb{E}[R_T]$ and high-probability guarantees that hold with probability $1 - \delta$, noting the latter can easily be converted to the former (while at most affecting logarithmic terms) by setting $\delta = 1/T$.

3 Main Results

We now formally state our main results, which give upper and lower bounds on the minimax regret of stochastic linear bandits under 1-bit per-batch communication constraints. These bounds match up to logarithmic factors in broad scaling regimes described below. A notable consequence of our bounds is that a single bit of feedback per batch suffices to match near-optimal *unquantized, unbatched* regret even for batch sizes as large as $B = \widetilde{O}(\sqrt{T})$. This demonstrates that the communication bottleneck is less severe than one might expect. The proofs are given in Appendices C, D, and E.

3.1 Regret Lower Bound

We begin by stating our algorithm-independent minimax lower bound.

Theorem 1 (Communication-Based Lower Bound)**.** *For any dimension $d \geq 1$, horizon $T \geq 1$, batch size $B \geq 1$, reward noise variance $\sigma^2 \geq 0$, and integer $K \geq 16$, there exists an action set $\mathcal{A}$ of size $|\mathcal{A}| = K$ with $\max_{a \in \mathcal{A}} \|a\|_2 \leq 1$ such that for every algorithm operating in $M = \lceil T/B \rceil$ batches that receives only one bit of feedback per batch, the expected cumulative regret satisfies*[5]

$$\sup_{\|\theta\|_2 \leq 1} \mathbb{E}[R_T] = \Omega\left(\min\{T, B \min(d, \log|\mathcal{A}|)\}\right).$$

We informally outline the proof idea as follows: When $|\mathcal{A}| \leq \exp(\Theta(d))$, the $B \log|\mathcal{A}|$ term arises because we can form $|\mathcal{A}|$ "well-separated" unit-norm vectors using tools from coding theory. We require $\log_2|\mathcal{A}|$ bits to identify the correct one (corresponding to $B \log|\mathcal{A}|$ pulls), and before this is achieved, we incur constant regret per round. When $|\mathcal{A}| \geq \exp(\omega(d))$, we can still use this argument on a subset of size $\exp(\Theta(d))$, with the remaining actions being "dummy" suboptimal actions.

Crucially, this lower bound holds even in the completely noiseless setting ($\sigma = 0$): even with perfect reward observations, the 1-bit uplink constraint alone forces $\Omega(B \min\{d, \log|\mathcal{A}|\})$ regret. This demonstrates that the bottleneck is purely a communication bottleneck rather than a statistical one.

For the remainder of the paper we assume $\sigma = \Theta(1)$ (see Remark 4 for extensions to other scalings); stochastic noise then introduces an additional statistical barrier on top of the communication bottleneck. Combining Theorem 1 with the well-known minimax lower bounds for standard linear bandits without communication constraints (e.g., (Zhou, 2019, Theorem 2 & Corollary 3)), we immediately obtain the following general lower bound.[6]

Corollary 1 (General Regret Lower Bound)**.** *Under the setup in Theorem 1 with $\sigma = \Theta(1)$, for any specified set size $|\mathcal{A}| \geq d$, there exists a corresponding action set $\mathcal{A}$ of that size with $\max_{a \in \mathcal{A}} \|a\|_2 \leq 1$ such that the expected cumulative regret satisfies*

$$\sup_{\|\theta\|_2 \leq 1} \mathbb{E}[R_T] = \widetilde{\Omega}\left(\min\left\{T,\ B \min\{d, \log|\mathcal{A}|\} + \sqrt{dT \min\{d, \log|\mathcal{A}|\}}\right\}\right).$$

When $|\mathcal{A}| = \exp(\Omega(d))$ this simplifies to $\widetilde{\Omega}\big(\min\{T,\ dB + d\sqrt{T}\}\big)$, and when $|\mathcal{A}| = \exp(O(d))$ this simplifies to $\widetilde{\Omega}\big(\min\{T,\ B \log|\mathcal{A}| + \sqrt{dT \log|\mathcal{A}|}\}\big)$. Here $\widetilde{\Omega}(\cdot)$ hides possible logarithmic dependencies on d in the final term.

The lower bound can be further tightened for the class of "non-mixing" algorithms, where in each batch $m = 1, \ldots, M$ the learner selects a *single* arm $A_m \in \mathcal{A}$, and the agent pulls this arm B times, i.e., every arm chosen in the batch must be identical.

[5] By defining the action set such that $\max_{a \in \mathcal{A}} \|a\|_2 \leq 1$ and allowing the adversary to pick any θ such that $\|\theta\|_2 \leq 1$, the mean reward satisfies $|\langle a, \theta \rangle| \leq \|a\|_2 \|\theta\|_2 \leq 1$ by the Cauchy-Schwarz inequality.

[6] The two lower bounds may use different action sets of the prescribed cardinality; taking their maximum suffices.

Theorem 2 (Lower Bound for Non-Mixing Algorithms)**.** *Fix a dimension $d \geq 2$, horizon $T \geq 1$, batch size $B \geq 1$, and reward noise variance $\sigma^2 \geq 0$. Let the action set be the standard basis $\mathcal{A} = \{e_1, \ldots, e_d\}$. For every non-mixing algorithm (as defined above), we have*

$$\sup_{\theta \in \mathcal{A}} \mathbb{E}[R_T] = \Omega(\min\{T, dB\}).$$

3.2 Regret Upper Bounds

With the lower bounds on regret established, we now give two constructive upper bounds: Theorem 3 gives a regret bound that is independent of the action set size $|\mathcal{A}|$, while Theorem 4 gives an $|\mathcal{A}|$-dependent upper bound that we then specialize to polynomial-size action sets. Section 4 explains the algorithms and their main ideas.

Theorem 3 ($|\mathcal{A}|$-Independent Upper Bound)**.** *Fix a finite action set $\mathcal{A} \subset \mathbb{R}^d$, and consider the problem described in Section 2 with time horizon $T \in \mathbb{N}$, variance $\sigma^2 = \Theta(1)$, and batch size $B \in \{1, \ldots, T\}$. Then, for any $\delta \in (0, 1)$, there exists an algorithm (see Algorithm 1 in Appendix D) that, with probability at least $1 - \delta$, incurs a regret of at most*

$$R_T = \widetilde{O}\left(dB \log\left(\frac{1}{\delta}\right) + d\sqrt{T \log\left(\frac{1}{\delta}\right)}\right), \tag{3}$$

where $\widetilde{O}$ hides logarithmic dependencies on d and T.

In view of Corollary 1, we see that by setting $\delta = 1/T$, Theorem 3 yields an expected regret that matches the minimax lower bound (up to logarithmic factors) when $|\mathcal{A}| = \exp(\Omega(d))$.

Remark 3 (Extension to Infinite Action Sets). Theorem 3 extends to any nonempty compact action set $\mathcal{A} \subset \mathbb{R}^d$ by discretization. Taking $\varepsilon = 1/T$ and $\mathcal{L} = \{\vartheta \in \mathbb{R}^d : \sup_{a \in \mathcal{A}} |\langle a, \vartheta \rangle| \leq 1\}$ in (Lattimore and Szepesvári, 2020, Exercise 27.6(c)), we obtain a finite subset $\mathcal{A}_{1/T} \subseteq \mathcal{A}$ with $|\mathcal{A}_{1/T}| \leq (6dT)^d$, such that every $a \in \mathcal{A}$ admits a representative $b \in \mathcal{A}_{1/T}$ with $\sup_{\vartheta \in \mathcal{L}} |\langle a - b, \vartheta \rangle| \leq 1/T$. Since $\theta \in \mathcal{L}$, the best mean reward on $\mathcal{A}_{1/T}$ is within $1/T$ of that on $\mathcal{A}$. Applying Theorem 3 to this finite set, working in its span if necessary, adds at most 1 to the regret relative to $\mathcal{A}$.

Theorem 4 ($|\mathcal{A}|$-Dependent Upper Bound)**.** *Fix a finite action set $\mathcal{A} \subset \mathbb{R}^d$. For the problem described in Section 2 with time horizon $T \in \mathbb{N}$, variance $\sigma^2 = \Theta(1)$, and batch size $B \in \{1, \ldots, T\}$, there exists an algorithm (see Algorithm 2 in Appendix E) that, with probability at least $1 - \delta$, incurs a regret of at most*

$$R_T = \widetilde{O}\left(B \log\left(\frac{|\mathcal{A}|}{\delta}\right) + d^{\frac{3}{2}}\sqrt{B} \log^{\frac{5}{2}}\left(\frac{|\mathcal{A}|}{\delta}\right) + \sqrt{dT \log\left(\frac{|\mathcal{A}|}{\delta}\right)}\right), \tag{4}$$

where $\widetilde{O}$ hides logarithmic dependencies on d and T.[7]

To better understand the behavior of this regret bound, we specialize Theorem 4 to the important special case where the action set size is polynomial in d. Here we set $\delta = 1/T$ to obtain a bound on the expected regret, though a high-probability counterpart naturally also follows.

Corollary 2 (Upper Bound for Polynomial-Size Action Sets)**.** *When $|\mathcal{A}| = \mathrm{poly}(d)$ and $\sigma = \Theta(1)$, the expected regret of Algorithm 2 is bounded by*

$$\mathbb{E}[R_T] = \widetilde{O}\left(B + d^{\frac{3}{2}}\sqrt{B} + \sqrt{dT}\right). \tag{5}$$

Notice that (5) consists of three components: the communication term B, the statistical term $\sqrt{dT}$, and an "algorithmic initialization penalty" $d^{3/2}\sqrt{B}$ (see Section 4 for discussion). The algorithm achieves near minimax optimality whenever this penalty term is absorbed by either of the two lower bounds. This is formalized in the following corollary, which also establishes that the degradation is not too severe when the penalty is not fully absorbed.

[7] The first two terms can be tightened to $\min\{B \log(|\mathcal{A}|/\delta) + d^{3/2}\sqrt{B} \log^{5/2}(|\mathcal{A}|/\delta),\ dB \log(|\mathcal{A}|/\delta)\}$ by taking the better of the small-B and large-B analyses, see Theorem 19 in Appendix E.

Corollary 3. *Assume $d \le |\mathcal{A}| = \mathrm{poly}(d)$ and $\sigma = \Theta(1)$. Write $\mathrm{LB}(B,T,d) \coloneqq \min\{T, B+\sqrt{dT}\}$. Compared to the minimax lower bound $\widetilde{\Omega}(\mathrm{LB}(B,T,d))$ from Corollary 1, the expected regret $\mathbb{E}[R_T]$ of Algorithm 2 is minimax order-optimal up to $\mathrm{polylog}(d,T)$ factors provided that at least one of $B = \Omega(d^3)$ or $T = \Omega(d^2 B)$ holds (which is guaranteed when $T = \Omega(d^5)$). Furthermore, in all remaining regimes, we have $\frac{\mathbb{E}[R_T]}{\mathrm{LB}(B,T,d)} \le \widetilde{O}(\sqrt{d})$.*

The proof of Corollary 3 is deferred to Appendix E.4. The condition $T/B = \Omega(d^2)$ requires that the horizon is long enough for the learner to receive $\Omega(d^2)$ batches of feedback; this is natural given that the agent transmits only 1 bit per batch, so the learner is severely information-starved in short horizons. However, Corollary 3 guarantees a gap of at most $\widetilde{O}(\sqrt{d})$ even in these harder regimes.

Remark 4 (Scaling of σ). For ease of presentation, and since it is the most widely-considered regime, we follow the convention in (Lattimore and Szepesvári, 2020, Chapter 19) and take $\sigma = \Theta(1)$ in Theorems 3 and 4 and their proofs. However, our methods readily extend to any σ that is not pathologically small. For any known σ such that $\sigma = O(1)$ and $\sigma = \Omega(T^{-c})$ for some fixed $c > 0$, a similar analysis can be performed using the same algorithms. By tracking the σ dependence in our analysis, the first term in (3) and the first two terms in (4) change by at most an additional $O\big(\log(1/\sigma)\big) = O(\log T)$ factor, while a multiplicative factor of σ appears in front of the final term in both equations. The final term in the lower bound also becomes $\widetilde{\Omega}(\min(T, \sigma\sqrt{dT\min\{d, \log|\mathcal{A}|\}}))$ (e.g., by slightly adapting the proof of (Tajdini et al., 2025, Thm. 2)), and near-optimality is preserved in broad scaling regimes.

Remark 5 (Multiple Bits per Batch). Our results can be generalized to the q-bit setting, where the agent may transmit $q \ge 1$ bits at the end of each batch. We outline the results here, and provide more details in Appendix G. The communication-based lower bound in Theorem 1 becomes $\Omega\big(\min\{T,\ B + (B/q)\min\{d, \log|\mathcal{A}|\}\}\big)$ with a very similar proof as the 1-bit case. The additive B term reflects the batching penalty that persists even with unquantized feedback, since the first batch's actions must be chosen before receiving any rewards. For the upper bounds, we define $\bar{q} = \min\{q, B\}$, since the benefit of having more bits turns out to saturate at B (at least for minimax regret in the $\widetilde{O}(\cdot)$ sense). Modified versions of Algorithms 1 and 2 attain the regret bounds in (3) and (4) with B replaced by $B/\bar{q}$ and an additional $\widetilde{O}(B\log(1/\delta))$ term. These bounds remain optimal up to logarithmic factors for exponentially large action sets, and for polynomial-size action sets whenever the resulting $d^{3/2}\sqrt{B/\bar{q}}$ term is dominated by the other terms.[8]

3.3 Discussion and Open Problems

While Theorem 3 attains near-optimal regret for exponential-sized action sets without any assumption on T, some gaps remain in the guarantees of Theorem 4 for smaller action sets. For $|\mathcal{A}| = \mathrm{poly}(d)$, Corollary 3 establishes that $T = \Omega(d^5)$ suffices for near order-optimality (or more generally, at least one of $B = \Omega(d^3)$ or $T = \Omega(d^2 B)$), which captures broad scalings but is stricter than the $T = \Omega(d)$ condition required in the standard unconstrained setting. For intermediate-sized action sets $|\mathcal{A}| = \exp(\Theta(d^\alpha))$ with $\alpha \in (0,1)$, a similar argument to Corollary 3 reveals that Theorem 4 achieves the target bound $\widetilde{O}(d^\alpha B + \sqrt{d^{1+\alpha}T})$ when either $B = \Omega(d^{3(1+\alpha)})$ or $T/B = \Omega(d^{2(1+2\alpha)})$. In summary, our results demonstrate near-optimality across broad scaling regimes of $(|\mathcal{A}|, B, d, T)$; closing the remaining gaps is an interesting direction for future work.

4 Proposed Algorithms

In this section, we introduce the two algorithms establishing Theorems 3 and 4. Both algorithms use a well-known *phased elimination* framework along with suitably-chosen *1-bit mean estimation* subroutines, but they have some important differences:

- The algorithm for Theorem 3 is non-mixing and favors simplicity, but it incurs a $\sqrt{d}$ factor in the confidence width that ultimately limits the regimes in which the regret bound is tight.

[8] The tightened bound given in the footnote to Theorem 4 also applies with B replaced by $B/\bar{q}$. In particular, for $q \ge B$, the small-batch bound, together with the additional batching overhead and the trivial regret bound, gives $\mathbb{E}[R_T] = \widetilde{O}(\min\{T, B+\sqrt{dT}\})$ for $d \le |\mathcal{A}| = \mathrm{poly}(d)$ and $\delta = 1/T$. Thus optimality up to logarithmic factors also holds in this regime.

- The algorithm for Theorem 4 modifies some of the key steps around arm allocation and robust mean estimation. For large B, it also crucially adds two *safe-arm identification* and *warm-start* subroutines in which mixing strategies are used. The motivation for these will be given below.

See Remark 9 in Appendix E for further comparison of the two algorithms, and Algorithms 1 and 2 in Appendices D and E for the full algorithm descriptions.

4.1 Preliminaries: Phased Elimination and 1-bit Mean Estimation

Phased elimination with G-optimal designs. Phased elimination algorithms (Lattimore et al., 2020; Camilleri et al., 2021; Bogunovic et al., 2021; Liu et al., 2024; Tajdini et al., 2025; Lattimore and Szepesvári, 2020) partition the time horizon into *epochs* $h = 1, 2, \dots$, maintaining an *active set* $\mathcal{A}_h \subseteq \mathcal{A}$ of plausibly-optimal arms (with $\mathcal{A}_1 = \mathcal{A}$). Within each epoch, the learner uses an *experimental design* $\rho_h \colon \mathcal{A}_h \to [0, 1]$ to efficiently estimate the mean rewards of *all* arms in $\mathcal{A}_h$ before eliminating those that appear suboptimal based on the confidence width W_h.

Throughout, we make use of near G-optimal designs; specifically, provided that $\mathcal{A}_h$ is compact and spans $\mathbb{R}^d$, there exist designs with the following guarantees:

$$\max_{a \in \mathcal{A}_h} \|a\|^2_{G_h^{-1}} \le 2d, \qquad |\mathrm{supp}(\rho_h)| \le 4d(\log\log d + 11) = \widetilde{O}(d), \tag{6}$$

where $G_h = \sum_{a \in \mathcal{A}_h} \rho_h(a) a a^\top$ and $\|a\|_V = \sqrt{a^\top V a}$ for a positive semi-definite matrix V. For explicitly represented finite active sets, such designs can be computed in polynomial time via the Frank-Wolfe algorithm (Todd, 2016, Proposition 3.17). The set $\mathrm{supp}(\rho_h)$ is called the *core set*, and notably has a size of only $\widetilde{O}(d)$. In the standard (unquantized and unbatched) setting, the rewards are directly observed, each core-set arm is pulled in proportion to ρ_h, and a weighted least-squares estimator $\hat{\theta}_h$ is formed. The mean reward of any arm $b \in \mathcal{A}_h$ is then estimated by $\langle b, \hat{\theta}_h \rangle$. With appropriate epoch lengths and elimination thresholds, this yields the standard minimax regret bound of $\widetilde{O}(\sqrt{dT \log|\mathcal{A}|})$.

Remark 6 (Active span). The properties in (6) are stated assuming that $\mathcal{A}_h$ spans $\mathbb{R}^d$. Although the initial action set satisfies this assumption by the convention in Remark 1, elimination may reduce the active span. We then work in the smaller subspace $\mathrm{span}(\mathcal{A}_h)$, interpreting G_h^{-1} as the inverse on this span. Since $\dim \mathrm{span}(\mathcal{A}_h) \le d$, the bounds in (6) remain valid with the initial dimension d; see (Lattimore et al., 2020, Remark 5.2).

Remark 7 (Computational complexity). As mentioned above, the core sets in each epoch h can be found in polynomial time using the Frank-Wolfe algorithm. However, the overall computational complexity in each epoch is $\Omega(|\mathcal{A}_h|)$ for a direct implementation, since estimating the rewards of the arms in $\mathcal{A}_h$ involves enumerating all $|\mathcal{A}_h|$ arms. This is potentially expensive in some scenarios (e.g., when $|\mathcal{A}| = \exp(\Omega(d))$), but this limitation is common to all phased elimination algorithms in which the arms are enumerated (e.g., see (Camilleri et al., 2021)). For suitably structured action sets, more efficient oracle-based implementations may be possible (see (Wagenmaker et al., 2021) for related results in the combinatorial semi-bandit setting), but we do not pursue this idea in this paper.

Adapting to our setting. Two immediate challenges arise in our constrained setup: (i) the learner observes only 1-bit summaries of B rewards rather than B unquantized rewards, rendering the standard weighted least-squares estimator unusable, and (ii) the allocation dictated by the continuous experimental design may not yield a whole number of batches, complicating the protocol.

We resolve (ii) by rounding per-arm allocations up to multiples of B: each batch then pulls either a single arm (in Algorithm 1, and in most batches of Algorithm 2) or one core-set arm plus a pre-determined *safe arm* to fill the remaining slots (in the warm-start of Algorithm 2).

To resolve (i), we replace the standard mean estimator with a *1-bit mean estimator* built on the framework of (Lau and Scarlett, 2026b,c). In our setting, a single 1-bit "query" is executed by having the agent pull the same arm B times (or possibly fewer in Algorithm 2, but we defer this distinction to later) and applying a 1-bit threshold to the sample mean. This reduces the effective standard deviation of the continuous measurement to $\sigma' = \sigma/\sqrt{B}$. With this established, we can employ two subroutines: 1BitLocalize uses $\widetilde{O}\left(\log \frac{1}{\sigma'} + \log \frac{1}{\delta}\right)$ queries to return an interval $[L_a, U_a] \subseteq [-1, 1]$ of width $O(\sigma')$ that contains μ_a with high probability; and 1BitRefine uses n_{q} additional queries (for some

suitably chosen n_{q}) to return a final estimate $\hat{Y}_{a,h}$ with variance $\widetilde{O}(\sigma'^2/n_{\mathrm{q}})$ and bias $O(\sigma'/T^2)$. The detailed subroutines and their formal guarantees are given in Appendix B.

Common template. Combining the preceding ingredients, a given epoch (indexed by h) in both algorithms proceeds as follows:

(1) Compute a near G-optimal design ρ_h on $\mathcal{A}_h$ satisfying (6).

(2) Run 1BitLocalize on each $a \in \mathrm{supp}(\rho_h)$ to obtain an interval $[L_a, U_a]$.

(3) Split the refinement budget into K blocks. For each block $j = 1, \ldots, K$ and each core-set arm $a \in \mathrm{supp}(\rho_h)$, run 1BitRefine to obtain independent scalar estimates $\hat{Y}_{a,h}^{(j)}$.

(4) Form a robust mean reward estimate $\hat{\mu}_{b,h}$ for every active arm $b \in \mathcal{A}_h$ using the scalar estimates $\hat{Y}_{a,h}^{(j)}$ and the design weights ρ_h. This involves computing design-weighted parameter estimates (using the inverse matrix G_h^{-1}) and applying median-of-means aggregation. The order of steps is algorithm-dependent: Algorithm 1 takes the median of the scalars first, whereas Algorithm 2 computes block-wise parameters first.

(5) Eliminate arms whose empirical gap exceeds twice the confidence width W_h:

$$\mathcal{A}_{h+1} = \left\{ a \in \mathcal{A}_h : \max_{b \in \mathcal{A}_h} \hat{\mu}_{b,h} - \hat{\mu}_{a,h} \le 2W_h \right\}.$$

The two algorithms differ in (a) how refinement samples are allocated across core-set arms, (b) the order of operations in Step 4, (c) the number of blocks K, (d) the confidence width W_h, and (e) additional pre-processing before the main loop (for the $|\mathcal{A}|$-dependent algorithm with large B). Both algorithms make use of a common precision parameter chosen as $\varepsilon_h = \sigma 2^{-h}/\sqrt{B}$, but they differ in how this translates to the final confidence width W_h. We now outline both algorithms in more detail.

4.2 Algorithm 1: $|\mathcal{A}|$-Independent Regret

For our $|\mathcal{A}|$-independent result (which is primarily suited to large action sets, e.g., exponential-sized), we exploit the observation that it suffices to accurately estimate only the $\widetilde{O}(d)$ scalars $\{\mu_a\}_{a \in \mathrm{supp}(\rho_h)}$ rather than all $|\mathcal{A}|$ arm means simultaneously, replacing a union bound over $|\mathcal{A}|$ arms with one over only the $\widetilde{O}(d)$ core-set arms. This enables two simplifications:

- Refinement samples are distributed *uniformly* across core-set arms (rather than proportionally to ρ_h), ensuring that each scalar block estimate $\hat{Y}_{a,h}^{(j)}$ has variance $O(\varepsilon_h^2)$ for valid localization intervals, with a bound independent of ρ_h.
- Median-of-means is applied directly to the scalar estimates to form a robust target $\hat{Y}_{a,h} = \mathrm{median}\{\hat{Y}_{a,h}^{(j)}\}_{j=1}^{K}$. The global parameter is then estimated via the design weights as

$$\hat{\theta}_h = G_h^{-1} \sum_{a \in \mathrm{supp}(\rho_h)} \rho_h(a) \cdot a \cdot \hat{Y}_{a,h} \qquad \text{where} \qquad G_h = \sum_{a \in \mathrm{supp}(\rho_h)} \rho_h(a)\, aa^\top. \tag{7}$$

  Because concentration is only required over the $\widetilde{O}(d)$ support arms, this requires only $K = \widetilde{O}(\log(d/\delta))$ blocks[9] rather than $\widetilde{O}(\log(|\mathcal{A}|/\delta))$.

The price paid for this simplification is a slightly looser confidence width $W_h = \sqrt{2d}\,\varepsilon_h$, arising from bounding the error propagation through $\hat{\theta}_h$ (via Corollary 4 in Appendix A). Pseudocode is given in Algorithm 1 in Appendix D, and the full analysis is also given in Appendix D.

4.3 Algorithm 2: $|\mathcal{A}|$-Dependent Regret

For smaller (e.g., polynomial-sized) action sets, the extra $\sqrt{d}$ penalty in the confidence width of Algorithm 1 prevents us from reaching minimax optimality. Algorithm 2 (detailed in Appendix E)

[9] When referring to the number of median-of-means blocks, we slightly abuse notation throughout and hide $\mathrm{poly}(\log T)$ factors in the $\widetilde{O}(\cdot)$ notation for ease of presentation.

rectifies this by allocating refinement samples *proportionally to* ρ_h (see (6)). By doing so, the resulting variances $\mathrm{Var}(\hat{Y}_{a,h}^{(j)}) \propto 1/\rho_h(a)$ perfectly cancel against the design weights when propagating through the block-wise parameter estimates

$$\hat{\theta}_h^{(j)} = G_h^{-1} \sum_{a \in \mathrm{supp}(\rho_h)} \rho_h(a) \cdot a \cdot \hat{Y}_{a,h}^{(j)} \qquad \text{where} \qquad G_h = \sum_{a \in \mathrm{supp}(\rho_h)} \rho_h(a)\, aa^\top, \tag{8}$$

yielding the tighter confidence width $W_h = \varepsilon_h$. However, because median-of-means must now be applied to the inner products $\hat{\mu}_{b,h} = \mathrm{median}\{\langle b, \hat{\theta}_h^{(j)} \rangle\}_{j=1}^K$ simultaneously for all $b \in \mathcal{A}_h$, ensuring concentration requires a union bound over the active set. Under our finite-variance assumption, this necessitates using $K = \widetilde{O}(\log(|\mathcal{A}|/\delta))$ blocks, contributing the $\sqrt{\log|\mathcal{A}|}$ factor to the regret.

Small-B regime. Let $\tau = \widetilde{O}(d \log(|\mathcal{A}|/\delta))$ denote the "threshold" between the small-B regime and large-B regime (see Line 3 of Algorithm 2 for a precise definition). When $B < \tau$, we simply initialize $\mathcal{A}_1 = \mathcal{A}$ and run the common template above for all epochs. While this incurs an initialization cost of $\widetilde{O}(dBK)$, the restricted size of B ensures this cost is bounded by $\widetilde{O}(d^{3/2}\sqrt{B}\log^{3/2}(|\mathcal{A}|/\delta))$.

Large-B regime. When $B \geq \tau$, an initialization term of $\widetilde{O}(dBK)$ becomes prohibitive, with the trivial global gap bound $\Delta_b \leq 2$ forcing significant over-exploration when $B \gg d$. We circumvent this by prepending two pre-processing steps that restrict the effective suboptimality gap of every arm entering $\mathcal{A}_1$ from $O(1)$ down to $O(\sqrt{d/B})$:

- *Safe arm identification* (Algorithm 3). Through at most $\lceil \log_2|\mathcal{A}| \rceil$ rounds of tournament-style bisection, we iteratively split the candidate set in half, using a single bit to indicate which half has the higher estimated maximum mean reward, and keep only the winning half. Within each round, the agent executes a G-optimal design *within a single batch*, pulling multiple core-set arms simultaneously and computing their unquantized sample means, in contrast to the single-arm batches used in the main phased elimination loop. After at most $\lceil \log_2|\mathcal{A}| \rceil$ rounds, the surviving arm a_{safe} has suboptimality gap at most $\widetilde{O}(\log^{3/2}(|\mathcal{A}|/\delta)\sqrt{d/B})$. This step uses $O(B \log|\mathcal{A}|)$ arm pulls.
- *Warm start* (Algorithm 4). We run a single warm-start epoch of phased elimination in which a_{safe} fills every slot not used by a core-set arm. This produces a refined active set $\mathcal{A}_1 \subseteq \mathcal{A}$ whose maximum gap is $O(\sqrt{d/B})$. The key insight is that a_{safe} absorbs the vast majority of slots in each batch, effectively decoupling the exploration budget from the rigid batch size constraint: core-set arms are pulled only $\widetilde{O}(B \log(|\mathcal{A}|/\delta))$ times in total. Together with the safe-arm gap bound, this yields the first two terms in (4), in place of the $\widetilde{O}(dBK)$ initialization regret incurred without warm starting.

Following these pre-processing steps, the main phased elimination loop proceeds on $\mathcal{A}_1$ according to the common template. Pseudocode for the main loop is given in Algorithm 2, with the pre-processing subroutines detailed in Algorithms 3 and 4. See Appendix E for the detailed regret analysis. While our problem formulation does not limit the amount of downlink communication, Appendix F summarizes what needs to be sent from the learner to the agent in each batch.

5 Conclusion

We have studied the problem of batched stochastic linear bandits with a 'batch-wise memoryless' agent and a 1-bit communication constraint. We presented two algorithms and provided a new minimax lower bound in this setting. For exponential-sized action sets, we provided an algorithm that matches our lower bound on the regret up to logarithmic factors, while for smaller action sets, we provided similar guarantees under broad scalings of the batch size, dimension, and time horizon. Notably, the regret from the standard setting can be matched to within logarithmic factors even in certain severely communication-constrained scenarios such as "1 bit per length-$\sqrt{T}$ batch". Perhaps the most immediate direction for future work is to devise an algorithm that attains near-optimal regret in the remaining scaling regimes where a gap remains.

Declaration of LLM Usage

The authors determined the overall research direction and led the development of the main results and proofs. Large language models (LLMs) were used to help draft and organize the submitted version of the manuscript. During the camera-ready revision, they also assisted with checking and refining technical arguments, including the conditioning arguments in Appendices D and E and the extension to multiple bits per batch in Appendix G. The authors reviewed and revised the LLM-assisted material and take full responsibility for the final manuscript and its results.

Acknowledgement

This work is supported by the Singapore National Research Foundation (NRF) and the Ministry of Digital Development and Information (MDDI) under the AI Visiting Professorship programme (Award AIVP-2024-003).

References

Abe, N. and Long, P. M. (1999). Associative reinforcement learning using linear probabilistic concepts. In *International Conference on Machine Learning*, pages 3–11. PMLR.

Amani, S., Lattimore, T., György, A., and Yang, L. (2023). Distributed contextual linear bandits with minimax optimal communication cost. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202, pages 691–717.

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

Bogunovic, I., Losalka, A., Krause, A., and Scarlett, J. (2021). Stochastic linear bandits robust to adversarial attacks. In *International Conference on Artificial Intelligence and Statistics*, pages 991–999. PMLR.

Bouneffouf, D., Bouzeghoub, A., and Gançarski, A. L. (2012). A contextual-bandit algorithm for mobile context-aware recommender system. In *International conference on neural information processing*, pages 324–331. Springer.

Camilleri, R., Jamieson, K., and Katz-Samuels, J. (2021). High-dimensional experimental design and kernel bandits. In *International Conference on Machine Learning*, pages 1227–1237. PMLR.

Cesa-Bianchi, N., Dekel, O., and Shamir, O. (2013). Online learning with switching costs and other adaptive adversaries. *Advances in Neural Information Processing Systems*, 26.

Chowdhury, S. R. and Zhou, X. (2022). Shuffle private linear contextual bandits. In *International Conference on Machine Learning*, pages 3984–4009. PMLR.

Chu, W., Li, L., Reyzin, L., and Schapire, R. (2011). Contextual bandits with linear payoff functions. In *International Conference on Artificial Intelligence and Statistics*, pages 208–214. JMLR.

Dani, V., Hayes, T. P., and Kakade, S. M. (2008). Stochastic linear optimization under bandit feedback. In *Annual Conference on Learning Theory (COLT)*, pages 355–366.

Ding, K., Li, J., and Liu, H. (2019). Interactive anomaly detection on attributed networks. In *ACM International Conference on Web Search and Data Mining*, pages 357–365.

Dubey, A. and Pentland, A. (2020). Differentially-private federated linear bandits. *Advances in Neural Information Processing Systems*, pages 6003–6014.

Durand, A., Achilleos, C., Iacovides, D., Strati, K., Mitsis, G. D., and Pineau, J. (2018). Contextual bandits for adapting treatment in a mouse model of de novo carcinogenesis. In *Machine Learning for Healthcare conference*, pages 67–82. PMLR.

Fan, L., Zhou, R., Tian, C., and Shen, C. (2023). Federated linear bandits with finite adversarial actions. *Advances in Neural Information Processing Systems*, pages 62549–62560.

Gao, Z., Han, Y., Ren, Z., and Zhou, Z. (2019). Batched multi-armed bandits problem. *Advances in Neural Information Processing Systems*, 32.

Gilbert, E. N. (1952). A comparison of signalling alphabets. *The Bell System Technical Journal*, 31(3):504–522.

Gretta, L. and Price, E. (2024). Sharp Noisy Binary Search with Monotonic Probabilities. In *51st International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 297, pages 75:1–75:19.

Hanna, O., Girgis, A. M., Fragouli, C., and Diggavi, S. (2024). Differentially private stochastic linear bandits: (almost) for free. *IEEE Journal on Selected Areas in Information Theory*, 5:135–147.

Hanna, O. A., Yang, L., and Fragouli, C. (2022). Solving multi-arm bandit using a few bits of communication. In *International Conference on Artificial Intelligence and Statistics*, pages 11215–11236. PMLR.

Hanna, O. A., Yang, L., and Fragouli, C. (2023). Contexts can be cheap: Solving stochastic contextual bandits with linear bandit algorithms. In *Conference on Learning Theory*, pages 1791–1821. PMLR.

He, J., Wang, T., Min, Y., and Gu, Q. (2022). A simple and provably efficient algorithm for asynchronous federated contextual linear bandits. *Advances in neural information processing systems*, 35:4762–4775.

Howson, B., Pike-Burke, C., and Filippi, S. (2023). Delayed feedback in generalised linear bandits revisited. In *International Conference on Artificial Intelligence and Statistics*, pages 6095–6119. PMLR.

Hu, J. and Zhong, H. (2026). Interaction is not necessary for order-optimal 1-bit mean estimation. *arXiv preprint arXiv:2608.02538*.

Huang, R., Wu, W., Yang, J., and Shen, C. (2021). Federated linear contextual bandits. *Advances in neural information processing systems*, 34:27057–27068.

Jun, K.-S., Jamieson, K., Nowak, R., and Zhu, X. (2016). Top arm identification in multi-armed bandits with batch arm pulls. In *Artificial Intelligence and Statistics*, pages 139–148. PMLR.

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

Lattimore, T., Szepesvari, C., and Weisz, G. (2020). Learning with good feature representations in bandits and in RL with a generative model. In *International Conference on Machine Learning*, pages 5662–5670. PMLR.

Lau, I. and Scarlett, J. (2025). Quantile multi-armed bandits with 1-bit feedback. In *Algorithmic Learning Theory*, pages 664–699. PMLR.

Lau, I. and Scarlett, J. (2026a). Non-adaptive 1-bit mean estimation: Minimax rates and the sample-interval tradeoff. *arXiv preprint arXiv:2609.08564*.

Lau, I. and Scarlett, J. (2026b). Order-optimal sequential 1-bit mean estimation in general tail regimes. *arXiv preprint arXiv:2604.07796*.

Lau, I. and Scarlett, J. (2026c). Sequential 1-bit mean estimation with near-optimal sample complexity. In *International Conference on Artificial Intelligence and Statistics*.

Li, C. and Wang, H. (2022). Asynchronous upper confidence bound algorithms for federated linear bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 6529–6553. PMLR.

Li, L., Chu, W., Langford, J., and Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *International Conference on World Wide Web*, pages 661–670.

Li, Y., Wang, Y., and Zhou, Y. (2019). Nearly minimax-optimal regret for linearly parameterized bandits. In *Conference on Learning Theory*, pages 2173–2174. PMLR.

Liu, H., Tajdini, A., Wagenmaker, A., and Wei, C.-Y. (2024). Corruption-robust linear bandits: Minimax optimality and gap-dependent misspecification. *Advances in Neural Information Processing Systems*, 37:24277–24325.

Lugosi, G. and Mendelson, S. (2019). Mean estimation and regression under heavy-tailed distributions: A survey. *Foundations of Computational Mathematics*, 19(5):1145–1190.

Luong, K., Son, D. T., Ta, H., Tran, H. T., and Dam, T. Q. (2026). Nearly optimal fixed-confidence best-arm identification with 1-bit feedback. *arXiv preprint arXiv:2610.02771*.

Mary, J., Gaudel, R., and Preux, P. (2015). Bandits and recommender systems. In *International Workshop on Machine Learning, Optimization and Big Data*, pages 325–336. Springer.

Mayekar, P., Scarlett, J., and Tan, V. Y. (2023). Communication-constrained bandits under additive Gaussian noise. In *International Conference on Machine Learning (ICML)*, pages 24236–24250.

Mehta, A., Saberi, A., Vazirani, U., and Vazirani, V. (2007). Adwords and generalized online matching. *Journal of the ACM (JACM)*, 54(5):22–es.

Miao, Y. (2026). Universal refinement without interaction: Order-optimal 1-bit mean estimation. *arXiv preprint arXiv:2607.24358*.

Mitra, A., Hassani, H., and Pappas, G. J. (2023). Linear stochastic bandits over a bit-constrained channel. In *Learning for Dynamics and Control Conference*, pages 1387–1399. PMLR.

Nava, E., Mutny, M., and Krause, A. (2022). Diversified sampling for batched Bayesian optimization with determinantal point processes. In *International Conference on Artificial Intelligence and Statistics*, pages 7031–7054. PMLR.

Polyanskiy, Y. and Wu, Y. (2025). *Information Theory: From Coding to Learning*. Cambridge University Press.

Provodin, D., Gajane, P., Pechenizkiy, M., and Kaptein, M. (2022). The impact of batch learning in stochastic linear bandits. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 1149–1154. IEEE.

Ren, X., Jin, T., and Xu, P. (2024). Optimal batched linear bandits. In *International Conference on Machine Learning*, pages 42391–42416. PMLR.

Ruan, Y., Yang, J., and Zhou, Y. (2021). Linear bandits with limited adaptivity and learning distributional optimal design. In *ACM SIGACT Symposium on Theory of Computing*, pages 74–87.

Salgia, S. and Zhao, Q. (2023). Distributed linear bandits under communication constraints. In *International Conference on Machine Learning*, pages 29845–29875. PMLR.

Shariff, R. and Sheffet, O. (2018). Differentially private contextual linear bandits. *Advances in Neural Information Processing Systems*, 31.

Shi, C., Zhou, R., Yang, K., and Shen, C. (2024). Harnessing the power of federated learning in federated contextual bandits. *Transactions on Machine Learning Research*.

Soemers, D., Brys, T., Driessens, K., Winands, M., and Nowé, A. (2018). Adapting to concept drift in credit card transaction data streams using contextual bandits and decision trees. In *AAAI Conference on Artificial Intelligence*, volume 32.

Tajdini, A., Scarlett, J., and Jamieson, K. (2025). Improved regret bounds for linear bandits with heavy-tailed rewards. *arXiv preprint arXiv:2506.04775*.

Todd, M. J. (2016). *Minimum-volume ellipsoids: Theory and algorithms*. SIAM.

Vakili, S., Ahmed, D., Bernacchia, A., and Pike-Burke, C. (2023). Delayed feedback in kernel bandits. In *International Conference on Machine Learning*, pages 34779–34792. PMLR.

Varshamov, R. R. (1957). Estimate of the number of signals in error correcting codes. *Docklady Akad. Nauk, SSSR*, 117:739–741.

Vernade, C., Carpentier, A., Lattimore, T., Zappella, G., Ermis, B., and Brueckner, M. (2020). Linear bandits with stochastic delayed feedback. In *International Conference on Machine Learning*, pages 9712–9721. PMLR.

Wagenmaker, A., Katz-Samuels, J., and Jamieson, K. (2021). Experimental design for regret minimization in linear bandits. In *International Conference on Artificial Intelligence and Statistics*. PMLR.

Zhang, Z., Ji, X., and Zhou, Y. (2025). Almost optimal batch-regret tradeoff for batch linear contextual bandits. In *International Conference on Learning Representations*.

Zhou, X. and Chowdhury, S. R. (2024). On differentially private federated linear contextual bandits. In *The Twelfth International Conference on Learning Representations*.

Zhou, Y. (2019). Lecture 14: Lower bounds for linear bandits. IE 498: Online Learning and Decision Making, University of Illinois at Urbana-Champaign. Lecture Notes.

A Concentration and Experimental Design Inequalities

Here we provide formal statements of several inequalities used throughout the paper. We begin by presenting the theoretical guarantee of the median of means estimator, with the precise statement being taken from (Lugosi and Mendelson, 2019, Theorem 2).

Lemma 5. *Let $X_1, \dots, X_n$ be a sequence of i.i.d. random variables each with mean μ and variance σ^2. Suppose that $n = mk$ for a positive integer m and $k = \lceil 8 \log(\frac{1}{\delta}) \rceil$. Then for any $\delta \in (0,1)$, the median-of-means estimator $\hat{\mu}_{\mathrm{MoM}}$ with k blocks satisfies*

$$|\hat{\mu}_{\mathrm{MoM}} - \mu| \le \sqrt{\frac{32\sigma^2}{n} \log\left(\frac{1}{\delta}\right)} \tag{9}$$

with probability at least $1 - \delta$.

The next two results concern experimental design properties that will be used throughout the subsequent proofs. While these results are well-known, we include their proofs for the sake of completeness.

Lemma 6. *Fix an arbitrary finite action set $\mathcal{X} \subset \mathbb{R}^d$ and let $\rho \colon \mathcal{X} \to [0,1]$ be a design satisfying (6) with respect to $\mathcal{X}$. For each action $b \in \mathcal{X}$, we have*

$$\sum_{a \in \mathrm{supp}(\rho)} \rho(a) \cdot (b^\top G^{-1} a)^2 \le 2d.$$

Proof. This follows from basic algebraic manipulations and the first property in (6):

$$\begin{aligned}
\sum_{a \in \mathrm{supp}(\rho)} \rho(a) \cdot (b^\top G^{-1} a)^2 &= \sum_{a \in \mathrm{supp}(\rho)} \rho(a) \cdot (b^\top G^{-1} a) \cdot (b^\top G^{-1} a) \\
&= \sum_{a \in \mathrm{supp}(\rho)} \rho(a) \cdot (b^\top G^{-1} a) \cdot (a^\top G^{-1} b) \\
&= b^\top G^{-1} \underbrace{\left(\sum_{a \in \mathrm{supp}(\rho)} \rho(a) a a^\top \right)}_{= G} G^{-1} b \\
&= b^\top G^{-1} b \\
&\le 2d
\end{aligned}$$

as desired. □

Corollary 4. *Under the same setup as in Lemma 6, we have for each action $b \in \mathcal{X}$ that*

$$\sum_{a \in \mathrm{supp}(\rho)} \rho(a) \cdot |b^\top G^{-1} a| \le \sqrt{2d}.$$

Proof. By the Cauchy-Schwarz inequality and $\sum_a \rho_h(a) = 1$, we have

$$\begin{aligned}
\sum_{a \in \mathrm{supp}(\rho)} \rho(a) \cdot |b^\top G^{-1} a| &= \sum_{a \in \mathrm{supp}(\rho)} \left(\sqrt{\rho(a)} \right) \cdot \left(\sqrt{\rho(a)} \cdot |b^\top G^{-1} a| \right) \\
&\le \sqrt{\sum_{a \in \mathrm{supp}(\rho)} \rho(a)} \cdot \sqrt{\sum_{a \in \mathrm{supp}(\rho)} \rho(a) \cdot (b^\top G^{-1} a)^2} \\
&\le \sqrt{2d},
\end{aligned}$$

where the last step follows from Lemma 6. □

B Details of 1-Bit Mean Estimation Subroutines

The subroutines in this section operate on a sequence of i.i.d. scalar observations $X_1, X_2, \ldots$ with mean $\mu := \mathbb{E}[X] \in [-1, 1]$ and variance $\mathrm{Var}(X) \le \sigma'^2$, where $\sigma' > 0$ is the *effective standard deviation* of a single observation.

The sample complexity is stated in terms of the number of 1-bit threshold *queries*: one query presents a single observation X_j to the agent, who returns the single bit $\mathbb{1}\{X_j \ge \gamma\}$ (or $\mathbb{1}\{X_j \le \gamma\}$) for a learner-chosen threshold γ. The cost in *raw arm pulls* equals the number of queries multiplied by the number of arm pulls per observation, which varies depending on the context. In our algorithms, two settings arise:

- **Full-batch queries** (Algorithm 1 and Step 3 of Algorithm 2): In each batch the agent pulls arm a exactly B times, computes the batch average $\bar{Y}_B = \frac{1}{B}\sum_{i=1}^{B} Y_i$, and returns a single bit. Thus
$$X = \bar{Y}_B, \qquad \sigma' = \frac{\sigma}{\sqrt{B}}, \qquad \text{and one query costs } B \text{ raw pulls of arm } a.$$
- **Partial-batch queries** (Algorithm 4, Warm-Start epoch): For arm $a \in \mathrm{supp}(\rho_0)$ with allocation $u_0(a) = \lceil B\rho_0(a) \rceil$, the agent pulls arm a exactly $u_0(a)$ times per batch (filling the remaining $B - u_0(a)$ slots with a_{safe}), computes the average of those $u_0(a)$ rewards, and returns a single bit. Thus
$$X = \bar{Y}_{u_0(a)}, \qquad \sigma' = \frac{\sigma}{\sqrt{u_0(a)}}, \qquad \text{and one query costs } u_0(a) \text{ raw pulls of arm } a.$$

Both subroutines below apply verbatim to either setting once σ' is identified from the context.[10]

B.1 1BitLocalize: Coarse Interval Estimation

When our bandit algorithms invoke $[L_a, U_a] \leftarrow \textsc{1BitLocalize}(a, \delta_{\mathrm{loc}}, \sigma')$, the subroutine executes the localization protocol of (Lau and Scarlett, 2026c) over $[-1, 1]$. This procedure treats each scalar observation X_j as a single noisy measurement of μ_a with standard deviation σ'.

Specifically, the protocol uses existing median estimation techniques such as noisy binary search (e.g., (Gretta and Price, 2024)) to first locate a high-probability confidence interval $[L'_a, U'_a]$ containing the median M of the observation clipped to $[-1, 1]$ using 1-bit threshold queries.[11] Then, by leveraging the well-known mean-median inequality $|\mathbb{E}[X] - M| \le \sigma'$, we naturally obtain a high-probability confidence interval $[L_a, U_a] = \big[\max\{-1, L'_a - \sigma'\}, \min\{1, U'_a + \sigma'\}\big]$ containing the true mean μ_a. See (Lau and Scarlett, 2026c, Appendix A, Step 1) for the full derivation leading to the following guarantee.

Proposition 7 (1BitLocalize guarantee)**.** *There exists an absolute constant $C_{\mathrm{loc}} > 0$ such that* $\textsc{1BitLocalize}(a, \delta_{\mathrm{loc}}, \sigma')$ *uses at most*
$$n_{\mathrm{loc}} = C_{\mathrm{loc}} \left(\log \frac{1}{\sigma'} + \log \frac{1}{\delta_{\mathrm{loc}}} \right)$$
queries[12] *and returns an interval $[L_a, U_a]$ satisfying*
$$\mathbb{P}\left(\mu_a \in [L_a, U_a] \ \text{ and } \ U_a - L_a \le 8\sigma'\right) \ge 1 - \delta_{\mathrm{loc}}.$$
The corresponding raw sample cost (i.e., total number of arm pulls) is n_{loc} multiplied by the number of raw arm pulls per query.

[10] Recent work (Lau and Scarlett, 2026a; Miao, 2026; Hu and Zhong, 2026) attains the order-optimal 1-bit mean-estimation rates with all localization and refinement queries chosen together in advance (i.e., non-adaptive queries), using general 1-bit queries instead of simple threshold queries. We retain the simpler adaptive threshold-query construction below, since adaptation across batches is permitted in our model.

[11] Clipping does not increase the variance or the distance of a median from $\mu_a \in [-1, 1]$.

[12] For ease of presentation, we assume σ' is strictly less than 1 throughout, so that the $\log(1/\sigma')$ term is non-negative. If $\sigma' \ge 1$, we can simply skip the localization step and return $[L_a, U_a] = [-1, 1]$, since $|[-1, 1]| \le 2\sigma'$ in this case.

B.2 1BITREFINE: Refined Mean Estimation

Fix the complete history immediately before refinement, including its inputs, and suppose that $\mu_a \in [L_a, U_a]$ and $U_a - L_a \leq 8\sigma'$. All probabilities, expectations, variances, and biases in this subsection are taken under the conditional law given this history. Refinement uses fresh i.i.d. observations and independent threshold randomization. Set $X' := X - (L_a + U_a)/2$. Then X' has mean $\mu' := \mu_a - (L_a + U_a)/2$, with $|\mu'| \leq 4\sigma'$ and $\mathrm{Var}(X') \leq \sigma'^2$. The base estimator proceeds as follows.

1. Modified Cutoff and Partitioning. Fix a target accuracy $\varepsilon > 0$. We select a truncation threshold

$$t = 2^{i_{\max}} \cdot \sigma', \quad \text{where} \quad i_{\max} := \left\lceil \log_2 T^2 \right\rceil, \tag{10}$$

so that $t \geq T^2\sigma'$. Following (Lau and Scarlett, 2026b), we then partition $[-t, t)$ into non-overlapping regions $R_1, R_{-1}, R_2, R_{-2}, \ldots, R_{i_{\max}}, R_{-i_{\max}}$ defined by exponentially growing interval boundaries $m_i\sigma'$:

$$R_i = \begin{cases} \sigma' \cdot [m_{i-1}, m_i) & \text{if } i \geq 1 \\ \sigma' \cdot \left[-m_{|i|}, -m_{|i|-1}\right) & \text{if } i \leq -1 \end{cases} \quad \text{where} \quad m_0 = 0 \text{ and } m_i = 2^i \text{ for } i \geq 1. \tag{11}$$

These half-open intervals assign every point of $[-t, t)$ to exactly one region.

Using a similar argument in (Lau and Scarlett, 2026b, Appendix A, Step 2), we can show that treating the insignificant region $\{X' \notin [-t, t)\}$ as zero introduces a truncation bias bounded by $O(\sigma'/T^2)$. We include the relevant steps here for completeness. For $T \geq 3$, since $|\mu'| \leq 4\sigma'$ under the localization event, we have

$$t \geq T^2\sigma' \geq 8\sigma' \geq 2|\mu'| \implies t - |\mu'| \geq \frac{t}{2} \implies \{|X'| \geq t\} \subseteq \{|X' - \mu'| \geq t/2\}.$$

Using this event inclusion together with the triangle inequality and Chebyshev's inequality, we obtain

$$\begin{aligned} |\mathbb{E}\left[X' \cdot \mathbb{1}\{X' \notin [-t, t)\}\right]| &\leq \mathbb{E}\left[|X' - \mu'| \cdot \mathbb{1}\left\{|X'| \geq t\right\}\right] + \mathbb{E}\left[|\mu'| \cdot \mathbb{1}\left\{|X'| \geq t\right\}\right] \\ &\leq \mathbb{E}\left[|X' - \mu'| \cdot \mathbb{1}\left\{|X' - \mu'| \geq \frac{t}{2}\right\}\right] + |\mu'| \cdot \mathbb{P}(|X'| \geq t) \\ &\leq \mathbb{E}\left[\frac{|X' - \mu'|^2}{t/2}\right] + |\mu'| \cdot \mathbb{P}(|X'| \geq t) \\ &\leq \frac{\sigma'^2}{t/2} + 4\sigma' \cdot \frac{\sigma'^2}{(t/2)^2} \\ &\leq \frac{2\sigma'}{T^2} + \frac{16\sigma'}{T^4} \\ &\leq \frac{3\sigma'}{T^2}, \end{aligned} \tag{12}$$

where the second last inequality follows from $t \geq T^2\sigma'$ and the last inequality holds for $T \geq 4$. We note that the choice of truncation threshold $t \geq T^2\sigma'$ makes the bias bound depend solely on T and σ', thus decoupling bias from the target accuracy parameter ε. This is the key modification we make relative to (Lau and Scarlett, 2026b), where the authors chose $t = \Theta(\sigma'^2/\varepsilon)$ to achieve a bias of $O(\varepsilon)$, which is sufficient for their purpose of 1-dimensional mean estimation.

2. Region-Wise Estimation via Threshold Queries. For each region $R_i = [a_i, b_i)$, we estimate its contribution $\mu_i' := \mathbb{E}[X' \cdot \mathbb{1}\{X' \in R_i\}]$ using the randomized threshold queries of (Lau and Scarlett, 2026b, Sec. 2.1, Step 4). Fix an absolute constant $C \geq 243$ and set $n = \lceil C\sigma'^2/\varepsilon^2 \rceil$, the number of observations for each query type. With $T_i \sim \mathrm{Unif}(a_i, b_i)$ independent of X', the four query types are

$$\mathbb{1}\{X' \geq a_i\}, \quad \mathbb{1}\{X' \geq T_i\}, \quad \mathbb{1}\{X' < b_i\}, \quad \mathbb{1}\{X' < T_i\}. \tag{13}$$

Each query type uses n fresh observations. Different query types and regions use disjoint observations, and every random-threshold query draws a fresh independent T_i. Once the input interval and parameters are fixed, all query thresholds can be chosen before any refinement responses are observed;

therefore the refinement stage is non-adaptive. Strict comparisons are obtained by complementing the corresponding non-strict threshold bits.[13]

Write $\hat{p}_{i,1}, \ldots, \hat{p}_{i,4}$ for the empirical means of the query responses in (13), in the displayed order. The local estimate for region R_i is

$$\hat{\mu}'_i = a_i(\hat{p}_{i,1} - \hat{p}_{i,2}) + b_i(\hat{p}_{i,3} - \hat{p}_{i,4})\,. \tag{14}$$

The final shifted estimate aggregates these local estimates across the entire collection of regions up to $i_{\max}$:

$$\hat{\mu}_a = \sum_{1 \le |i| \le i_{\max}} \hat{\mu}'_i + \frac{L_a + U_a}{2},$$

which satisfies the following guarantee.

Lemma 8 (1BitRefine fixed-accuracy)**.** *Let $T \ge 4$. Given inputs T, σ', ε as well as a valid interval $[L_a, U_a]$ containing μ_a, with $U_a - L_a \le 8\sigma'$, the estimator described above uses at most*

$$n_{\text{base}} = 8\left\lceil \frac{C\sigma'^2}{\varepsilon^2} \right\rceil \lceil 2\log_2 T \rceil = O\left(\left(1 + \frac{\sigma'^2}{\varepsilon^2}\right)\log T\right) \tag{15}$$

queries, and its output $\hat{\mu}_a$ satisfies

$$\mathrm{Var}(\hat{\mu}_a) \le \varepsilon^2 \qquad \textit{and} \qquad |\mathrm{Bias}(\hat{\mu}_a)| \le \frac{3\sigma'}{T^2}\,. \tag{16}$$

Proof. There are four query types in each of the $2i_{\max}$ regions, so the total query count is $4{\cdot}2i_{\max}{\cdot}n = n_{\text{base}}$, as in (15).

For a fixed realization $X' = x \in R_i$, averaging over $T_i \sim \mathrm{Unif}(a_i, b_i)$ gives

$$\mathbb{E}[\mathbb{1}\{x \ge a_i\} - \mathbb{1}\{x \ge T_i\}] = 1 - \mathbb{P}(T_i \le x) = \frac{b_i - x}{b_i - a_i}$$

and

$$\mathbb{E}[\mathbb{1}\{x < b_i\} - \mathbb{1}\{x < T_i\}] = 1 - \mathbb{P}(T_i > x) = \frac{x - a_i}{b_i - a_i}.$$

For $x \notin R_i$, the two indicators in each pair above coincide, so both differences vanish. Since each empirical mean is unbiased (i.e., its expectation is precisely the corresponding query-response probability), taking expectations in (14) yields the unbiasedness of the local estimates:

$$\mathbb{E}[\hat{\mu}'_i] = \mathbb{E}\left[\frac{a_i(b_i - X') + b_i(X' - a_i)}{b_i - a_i} \cdot \mathbb{1}\{X' \in R_i\}\right] = \mathbb{E}[X'\mathbb{1}\{X' \in R_i\}] = \mu'_i.$$

The regions $R_{\pm 1}, R_{\pm 2}, \ldots, R_{\pm i_{\max}}$ partition $[-t, t)$, so the bias of $\hat{\mu}_a$ is $-\mathbb{E}[X'\mathbb{1}\{X' \notin [-t, t)\}]$. The claimed bound on this bias in (16) follows directly from (12).

For the variance, we sum the independent contributions across the regions, following (Lau and Scarlett, 2026b, Appendix A, Step 5(a)–(b)). Since $\hat{p}_{i,1}, \ldots, \hat{p}_{i,4}$ are independent (due to relying on disjoint sets of samples), the variance expands as

$$\mathrm{Var}(\hat{\mu}'_i) = a_i^2\mathrm{Var}(\hat{p}_{i,1}) + a_i^2\mathrm{Var}(\hat{p}_{i,2}) + b_i^2\mathrm{Var}(\hat{p}_{i,3}) + b_i^2\mathrm{Var}(\hat{p}_{i,4}).$$

For the two central regions $i \in \{-1, 1\}$, we have $a_1 = b_{-1} = 0$ and $b_1 = -a_{-1} = 2\sigma'$. Since each empirical Bernoulli mean $\hat{p}_{i,k}$ has variance at most $1/n$, these regions contribute at most $16\sigma'^2/n$ to the variance.

For outer-right regions $i \ge 2$, a response of 1 to either of the first two queries in (13), or 0 to either of the last two, requires $X' \ge a_i$. Hence, the Bernoulli variance bound $p(1-p) \le \min\{p, 1-p\}$ gives $\mathrm{Var}(\hat{p}_{i,k}) \le \mathbb{P}(X' \ge a_i)/n$ for $k = 1, \ldots, 4$. Since $a_i = m_{i-1}\sigma'$ and $b_i = 2a_i$, we obtain

$$\mathrm{Var}(\hat{\mu}'_i) \le \frac{2a_i^2 + 2b_i^2}{n}\mathbb{P}(X' \ge a_i) = \frac{10m_{i-1}^2\sigma'^2}{n}\mathbb{P}(X' \ge m_{i-1}\sigma')\,.$$

[13] Alternatively, the interval queries $\mathbb{1}\{a_i \le X' < T_i\}$ and $\mathbb{1}\{T_i \le X' < b_i\}$, also permitted by our model, require only two query types per region instead of four; see (Lau and Scarlett, 2026c, Appendix A, Steps 4–5).

For the outer-left regions R_{-i} with $i \geq 2$, the same argument uses $\mathbb{P}(X' \leq -m_{i-1}\sigma')$, with the roles of responses 0 and 1 reversed. Adding these bounds yields

$$\mathrm{Var}(\hat{\mu}_i') + \mathrm{Var}(\hat{\mu}_{-i}') \leq \frac{10 m_{i-1}^2 \sigma'^2}{n} \mathbb{P}(|X'| \geq m_{i-1}\sigma'). \tag{17}$$

Summing (17) over all $i \geq 2$ and adding the central-region bound gives

$$\mathrm{Var}(\hat{\mu}_a) \leq \frac{16\sigma'^2}{n} + \frac{10}{n} \sum_{i=2}^{i_{\max}} m_{i-1}^2 \sigma'^2 \cdot \mathbb{P}(|X'| \geq m_{i-1}\sigma'). \tag{18}$$

We write

$$m_{i-1}^2 \sigma'^2 \cdot \mathbb{P}(|X'| \geq m_{i-1}\sigma') = \mathbb{E}[m_{i-1}^2 \sigma'^2 \cdot \mathbb{1}\{|X'| \geq m_{i-1}\sigma'\}],$$

and bound the summation in (18) pointwise for any fixed realization $X' = x$. For any active index in the sum, the condition $|x| \geq m_{i-1}\sigma'$ ensures that $m_{i-1}^2 \sigma'^2 \leq x^2$. Because the thresholds double ($m_i = 2m_{i-1}$), summing in reverse from the largest active index bounds the sum by a geometric series that decays by a factor of 4 at each step:

$$\sigma'^2 \sum_{i=2}^{i_{\max}} m_{i-1}^2 \mathbb{1}\{|x| \geq m_{i-1}\sigma'\} \leq x^2 \left(1 + \frac{1}{4} + \frac{1}{16} + \cdots\right) = \frac{4x^2}{3}. \tag{19}$$

Taking the expectation of (19) with respect to X' and combining with (18) yields

$$\mathrm{Var}(\hat{\mu}_a) \leq \frac{1}{n}\left(16\sigma'^2 + \frac{40}{3}\mathbb{E}[X'^2]\right) \leq \frac{1}{n}\left(16\sigma'^2 + \frac{40\left(\sigma'^2 + |\mu'|^2\right)}{3}\right) \leq \frac{C\sigma'^2}{n} \leq \varepsilon^2,$$

where the last two inequalities follow from $|\mu'| \leq 4\sigma'$, $16 + (40/3) \cdot 17 = 728/3 < C$, and $n \geq C\sigma'^2/\varepsilon^2$. □

In our phased elimination bandit algorithms, rather than specifying a target accuracy upfront, the learner allocates some fixed query budget n_{q} to each arm. By inverting the sample complexity bound in Lemma 8, we obtain the following fixed-budget guarantee, which serves as the primary formal interface invoked by our bandit algorithms via the call $\hat{Y}_{a,h}^{(j)} \leftarrow \textsc{1BitRefine}(a, [L_a, U_a], n_{\mathrm{q}})$

Corollary 5 (1BitRefine fixed-budget)**.** *Let $T \geq 4$, and let C be the constant from Lemma 8. Given a fixed input interval $[L_a, U_a]$ containing μ_a with $U_a - L_a \leq 8\sigma'$, suppose that $n_{\mathrm{q}} \geq 48 \log_2 T$ queries are available. Running the estimator above with target accuracy set to $\varepsilon = \sigma'\sqrt{(48C \log_2 T)/n_{\mathrm{q}}}$ yields an estimate $\hat{\mu}_a$ satisfying*

$$\mathrm{Var}(\hat{\mu}_a) \leq \frac{C_V \sigma'^2 \log_2 T}{n_{\mathrm{q}}} \qquad \textit{and} \qquad |\mathrm{Bias}(\hat{\mu}_a)| \leq \frac{3\sigma'}{T^2}, \tag{20}$$

where $C_V = 48C \geq 48$ is a universal absolute constant.

Proof. We verify that the available query budget n_{q} is sufficient to run the estimator in Lemma 8 at the chosen target accuracy ε, i.e., $n_{\mathrm{q}} \geq n_{\mathrm{base}}$. Since $n_{\mathrm{q}} \geq 48 \log_2 T$, we have $C\sigma'^2/\varepsilon^2 = n_{\mathrm{q}}/(48 \log_2 T) \geq 1$, so applying $\lceil x \rceil \leq 2x$ for $x \geq 1$, we get

$$n_{\mathrm{base}} \leq 8 \left\lceil \frac{C\sigma'^2}{\varepsilon^2} \right\rceil \lceil 2 \log_2 T \rceil \leq 8 \cdot \left(\frac{2n_{\mathrm{q}}}{48 \log_2 T}\right) \cdot 3 \log_2 T = n_{\mathrm{q}}.$$

Substituting ε into Lemma 8 gives $\mathrm{Var}(\hat{\mu}_a) \leq \varepsilon^2 = C_V \sigma'^2 \log_2 T / n_{\mathrm{q}}$, where $C_V = 48C \geq 48$. The bias bound follows directly from (12), which holds under the fixed valid-transcript conditioning, regardless of ε or n_{q}. Each completed call uses its full query allocation; any queries not needed by the estimator use fresh observations whose bits are simply discarded. □

C Proofs of Lower Bounds

C.1 Proof of Theorem 1 (General Lower Bound)

We first state a connection between Hamming distance and inner product, whose proof is immediate from their definitions (and thus omitted).

Lemma 9. *Let $x, x' \in \{\pm 1/\sqrt{d}\}^d$. If the Hamming distance $d_{\mathrm{H}}(x, x') \coloneqq \sum_{i=1}^{d} \mathbb{1}\{x_i \neq x'_i\}$ between x and x' is h, then*

$$\langle x, x' \rangle = 1 - \frac{2h}{d}.$$

In particular, if $h \geq d/4$, then $\langle x, x' \rangle \leq 1/2$ for each $x \neq x'$.

Therefore, in order to produce a set $\mathcal{V} \subset \{\pm 1/\sqrt{d}\}^d$ with small pairwise inner products, it suffices to construct a binary code $\mathcal{C} \subset \{0,1\}^d$ with block length d and Hamming distance at least $d/4$. The Gilbert–Varshamov (GV) bound guarantees the existence of such a code.

Proposition 10 (A specific instantiation of (Gilbert, 1952; Varshamov, 1957))**.** *For every integer d, there exists a binary code $\mathcal{C} \subset \{0,1\}^d$ with minimum Hamming distance at least $\lceil d/4 \rceil$ and size*

$$|\mathcal{C}| \geq 2^{(1 - H_2(1/4))d - o(d)},$$

where $H_2(p) = -p \log_2 p - (1-p)\log_2(1-p)$ is the binary entropy function. Numerically $1 - H_2(1/4) \approx 0.1887$, so $|\mathcal{C}| \geq 2^{cd}$ for a constant $c > 0$.

We now prove Theorem 1 using the bounds above.

Proof of Theorem 1. Since instantaneous regrets are non-negative, we have $R_T \geq R_{B\lfloor T/B \rfloor}$. Therefore, we may assume $T = BM$ for integer M without loss of generality, making every batch exactly B pulls. Fix the GV-rate constant $c > 0$ from Proposition 10 for $\delta = 1/4$. We consider two cases of d.

Case 1: $d < 4/c$ (small, constant dimension). We first consider $|\mathcal{A}| = 2$ and then generalize to higher $|\mathcal{A}|$. Consider $\mathcal{A} = \{e_1, -e_1\} \subset \mathbb{R}^d$ with e_1 being the first standard basis vector, and draw θ uniformly from $\mathcal{A}$. The optimal arm is θ (with mean reward $+1$) and the only other arm is $-\theta$ (with mean reward -1), so every suboptimal pull incurs regret 2. Before the first batch the learner has seen zero bits of feedback, and hence has no information about θ. By symmetry, every pull in batch 1 is suboptimal with probability $1/2$. Summing over the $\min\{T, B\}$ pulls in the first batch gives $\mathbb{E}[R_T] \geq \min\{T, B\} \cdot 2 \cdot \frac{1}{2} = \min\{T, B\}$. This matches the target lower bound when $|\mathcal{A}| = 2$, because $\min(d, \log|\mathcal{A}|) = O(1)$, and so $\min\{T, B\} = \Omega(\min\{T, B\min(d, \log|\mathcal{A}|)\})$.

When $2 \leq d < 4/c$ and $|\mathcal{A}| > 2$, a similar argument applies by adding a further $|\mathcal{A}| - 2$ "dummy" arms whose first coordinate is zero, thus meaning these arms are strictly suboptimal and their rewards convey no information about which of $\{e_1, -e_1\}$ is optimal. When $d = 1$ and $|\mathcal{A}| > 2$, a similar argument holds with the dummy arms having low value (e.g., in $[-1/2, 1/2]$); a similar idea will be used in Case 2 below, so is skipped here.

For the rest of the proof, we assume $d \geq 4/c$, thus guaranteeing $\lfloor 2^{cd} \rfloor \geq 16$.

Case 2: $d \geq 4/c$ (large dimension). To construct our hard instance for a given size $|\mathcal{A}| = K$, we first define the size of our active hard subset: $K' = \min\{K, \lfloor 2^{cd} \rfloor\}$. By Proposition 10, there exists a binary code $\mathcal{C} \subset \{0,1\}^d$ with minimum distance at least $\lceil d/4 \rceil$ and size at least 2^{cd}. We select an arbitrary subset of this code of size exactly K', and map this subset to $\mathcal{V} \subset \{\pm 1/\sqrt{d}\}^d$ via the bijection $0 \mapsto -1/\sqrt{d}$ and $1 \mapsto +1/\sqrt{d}$. Let this hard subset be $\mathcal{A}'$.

To expand this to the full size K, we add $K - K'$ arms that are distinct but suboptimal by at least $1/2$. Since $\mathcal{B}(1/2) = \{x \in \mathbb{R}^d : \|x\|_2 \leq 1/2\}$, i.e., the ball of radius $1/2$, contains uncountably many vectors, we can construct the remaining $K - K'$ actions by selecting arbitrary distinct vectors with an L_2 norm of at most $1/2$. Let this padding set be $\mathcal{A}_{\mathrm{pad}}$, and define our action set as $\mathcal{A} = \mathcal{A}' \cup \mathcal{A}_{\mathrm{pad}}$, so $|\mathcal{A}| = K$.

To lower bound the worst-case expected regret, we apply Yao's minimax principle and instead lower bound the expected regret of every deterministic algorithm when the true parameter Θ is drawn uniformly from the hard subset $\mathcal{A}'$. Note that $\|\Theta\|_2 = 1$.

Under this prior, the unique optimal arm is Θ, yielding a mean reward of $\|\Theta\|_2^2 = 1$. If the learner plays any suboptimal arm $a \in \mathcal{A}' \setminus \{\Theta\}$, the mean reward is at most $1/2$ by Lemma 9 and the Gilbert-Varshamov code construction. If the learner plays any padding arm $a \in \mathcal{A}_{\text{pad}}$, the mean reward is at most $1/2$ by the Cauchy-Schwarz inequality:

$$|\langle a, \Theta \rangle| \leq \|a\|_2 \|\Theta\|_2 \leq \frac{1}{2}(1) = \frac{1}{2}.$$

Thus, any arm pulled from $\mathcal{A}$ that is not exactly the true parameter Θ incurs an instantaneous regret of at least $\Delta = 1/2$.

Fix a batch index $i \in \{1, \ldots, M\}$. Let $\mathbf{Q}_{i-1} = (Q_1, \ldots, Q_{i-1})$ denote the sequence of 1-bit messages revealed after the first $i-1$ batches. By the 1-bit communication constraint, the history $\mathbf{Q}_{i-1}$ consists of exactly $i-1$ bits. Therefore, its entropy and hence its mutual information with Θ is trivially bounded by

$$I(\Theta; \mathbf{Q}_{i-1}) \leq H(\mathbf{Q}_{i-1}) \leq i - 1.$$

In batch i, the learner selects B actions $A_{i,1}, \ldots, A_{i,B}$. Because the algorithm cannot observe rewards mid-batch, every action $A_{i,t}$ is a deterministic function of the history $\mathbf{Q}_{i-1}$. Since Θ is drawn uniformly from K' possibilities, we can apply Fano's inequality (Polyanskiy and Wu, 2025, Corollary 3.13) to bound the probability that any specific pull successfully selects the optimal arm:

$$\mathbb{P}\big(A_{i,t} = \Theta\big) \leq \frac{I(\Theta; \mathbf{Q}_{i-1}) + \log_2 2}{\log_2 K'} \leq \frac{(i-1)+1}{\log_2 K'} = \frac{i}{\log_2 K'}.$$

Thus, the probability that a pull in batch i is suboptimal is at least $1 - \frac{i}{\log_2 K'}$. Since any suboptimal pull incurs a gap of at least $1/2$, the expected instantaneous regret for *every single pull* in batch i is at least

$$\frac{1}{2} \cdot \left(1 - \frac{i}{\log_2 K'}\right).$$

Let $L = \lfloor \log_2 K' \rfloor$. Because $K \geq 16$ and $\lfloor 2^{cd} \rfloor \geq 16$ (since $d \geq 4/c$), we have $K' \geq 16$, meaning $L \geq 4$. We consider two cases for the total number of batches $M = T/B$: (i) $M \geq L$, and (ii) $M < L$.

For case (i), the expected regret generated by the B pulls in each batch $i = 1, \ldots, L$ is at least $\frac{B}{2}\big(1 - \frac{i}{\log_2 K'}\big)$. Therefore, the expected regret over just the first L batches is

$$\mathbb{E}_{\Theta \sim \text{Unif}(\mathcal{A}')}[R_{LB}] \geq \sum_{i=1}^{L} \frac{B}{2}\Big(1 - \frac{i}{\log_2 K'}\Big) = \frac{B}{2}\Big(L - \frac{L(L+1)}{2\log_2 K'}\Big).$$

Because $L = \lfloor \log_2 K' \rfloor$, we know $\frac{L+1}{\log_2 K'} \leq \frac{\log_2 K' + 1}{\log_2 K'} = 1 + \frac{1}{\log_2 K'}$. Since $K' \geq 16$, this ratio is at most $1 + 1/4 = 1.25$, and we obtain

$$\mathbb{E}_{\Theta \sim \text{Unif}(\mathcal{A}')}[R_T] \geq \mathbb{E}_{\Theta \sim \text{Unif}(\mathcal{A}')}[R_{LB}] \geq \frac{B}{2}\Big(L - \frac{1.25L}{2}\Big) = \frac{B}{2}(0.375L) = \Omega(B \log K').$$

For case (ii), we sum the per-batch regret bound over all M batches, yielding

$$\mathbb{E}_{\Theta \sim \text{Unif}(\mathcal{A}')}[R_T] \geq \sum_{i=1}^{M} \frac{B}{2}\Big(1 - \frac{i}{\log_2 K'}\Big) = \frac{B}{2}\Big(M - \frac{M(M+1)}{2\log_2 K'}\Big).$$

Since M and L are integers with $M < L$, we have $M + 1 \leq L \leq \log_2 K'$. This implies $\frac{M+1}{\log_2 K'} \leq 1$. Consequently, we have

$$\mathbb{E}_{\Theta \sim \text{Unif}(\mathcal{A}')}[R_T] \geq \frac{B}{2}\Big(M - \frac{M}{2}\Big) = \frac{BM}{4} = \Omega(T).$$

Combining both cases, the expected regret is $\mathbb{E}_{\Theta \sim \text{Unif}(\mathcal{A}')}[R_T] = \Omega(\min\{T, B \log K'\})$. By Yao's minimax principle, the worst-case expected regret is lower bounded by this expected regret. Substituting $K' = \min\{K, \lfloor 2^{cd} \rfloor\}$ and noting that $\log \lfloor 2^{cd} \rfloor = \Theta(d)$, we conclude that

$$\sup_{\|\theta\|_2 \leq 1} \mathbb{E}[R_T] \geq \mathbb{E}_{\Theta \sim \text{Unif}(\mathcal{A}')}[R_T] = \Omega\left(\min\{T, B \min(d, \log|\mathcal{A}|)\}\right).$$

This completes the proof. □

C.2 Proof of Theorem 2 (Lower Bound for Non-Mixing Algorithms)

Proof of Theorem 2. To lower bound the worst-case expected regret, we again apply Yao's minimax principle, restrict attention to deterministic algorithms, and assume that the adversary draws the true parameter Θ uniformly from $\mathcal{A} = \{e_1, \ldots, e_d\}$. Under this prior, the unique optimal arm is Θ with a mean reward of 1, while any other arm $e_j \neq \Theta$ yields a mean reward of 0.

To emphasize the severity of the non-mixing constraint, we consider a strictly easier environment: We remove any noise (i.e., consider $\sigma = 0$) and allow the agent to transmit the exact, noiseless rewards to the learner, completely bypassing the 1-bit communication constraint. Any lower bound in this idealized setting trivially applies to the communication-constrained (and possibly noisy) problem.

In this noiseless setting, pulling (or "querying") a single arm $A \in \mathcal{A}$ for a batch yields B identical observations of $\mathbb{1}\{A = \Theta\}$. Because the vectors in $\mathcal{A}$ are orthogonal, pulling any $A \neq \Theta$ yields a reward of 0. Thus, a query only reveals whether $A = \Theta$ or $A \neq \Theta$. Until the optimal arm is successfully queried, the learner receives a sequence of identical 0s.

Let S_i be the set of arms queried over the first i batches. Because the algorithm is non-mixing, it can test at most one arm per batch, and thus $|S_i| \leq i$. Since the sequence of 0s provides no distinguishing information about the unqueried arms, the posterior distribution of Θ remains uniform over the remaining unqueried arms. Testing a previously unqueried arm in each batch therefore finds Θ within i batches with probability i/d for $i \leq d$, and repeated queries cannot increase this probability. Hence,

$$\mathbb{P}(\Theta \in S_i) \leq \frac{i}{d}.$$

Let $M_0 = \lfloor d/2 \rfloor$. For any number of batches $i \leq M_0$, the probability that the algorithm has *not* yet found the optimal arm is bounded by:

$$\mathbb{P}(\Theta \notin S_i) = 1 - \mathbb{P}(\Theta \in S_i) \geq 1 - \frac{i}{d} \geq 1 - \frac{\lfloor d/2 \rfloor}{d} \geq \frac{1}{2}.$$

If $\Theta \notin S_i$, the algorithm has failed to find the optimal arm in the first i batches. For any $i < \lceil T/B \rceil$, this corresponds to iB suboptimal pulls, accumulating exactly iB regret. Therefore, the expected regret over the first i batches is at least

$$\mathbb{E}_{\Theta \sim \mathrm{Unif}(\mathcal{A})}[R_{iB}] \geq \mathbb{P}(\Theta \notin S_i) \cdot (iB) \geq \frac{iB}{2}.$$

We compute the total expected regret $\mathbb{E}_\Theta[R_T]$ by considering two cases for the total number of batches $M = \lceil T/B \rceil$:

- **Case 1 ($M \leq M_0$):** The algorithm halts before reaching the threshold M_0. By evaluating the failure probability at the final batch M, the algorithm fails to find the optimal arm over all T pulls with probability at least $1 - M/d \geq 1 - M_0/d \geq 1/2$. Thus, the expected regret over the entire horizon is at least $\frac{1}{2}T = \Omega(T)$.
- **Case 2 ($M > M_0$):** The algorithm continues past the threshold. Since regret is strictly non-decreasing, we can lower bound the total regret by the expected regret accumulated during just the first M_0 batches. This yields at least $\frac{1}{2}M_0 B = \frac{1}{2}\lfloor d/2 \rfloor B = \Omega(dB)$.

Combining both cases, the expected regret is $\mathbb{E}_{\Theta \sim \mathrm{Unif}(\mathcal{A})}[R_T] = \Omega(\min\{T, dB\})$. By Yao's minimax principle, the worst-case expected regret is lower bounded by this expected regret:

$$\sup_{\theta \in \mathcal{A}} \mathbb{E}[R_T] \geq \mathbb{E}_{\Theta \sim \mathrm{Unif}(\mathcal{A})}[R_T] = \Omega(\min\{T, dB\}).$$

This completes the proof. □

D Proof of Theorem 3 ($|\mathcal{A}|$-Independent Upper Bound)

In this section, we formally establish Theorem 3 by providing the detailed pseudocode and regret analysis of our first algorithm with an $|\mathcal{A}|$-independent regret bound. As discussed in Section 4.1, this approach exploits the fact that it suffices to estimate only the $|\mathrm{supp}(\rho_h)| = \widetilde{O}(d)$ scalar quantities

$\{\langle a, \theta \rangle\}_{a \in \mathrm{supp}(\rho_h)}$ to successfully recover $\hat{\theta}_h$. Thus, we can apply median-of-means directly to these individual scalar estimates, limiting the required number of blocks to $K = \widetilde{O}(\log(d/\delta))$. Furthermore, by distributing the refinement queries uniformly across the core set, each scalar block estimate $\hat{Y}_{a,h}^{(j)}$ has variance at most $\varepsilon_h^2/16$ conditional on a fixed valid localization history. Median-of-means and Corollary 4 then give the confidence width $W_h = \sqrt{2d}\,\varepsilon_h$.

The detailed pseudocode is presented in Algorithm 1. We first proceed with the analysis by first bounding the sample complexity of the localization and refinement stages (Appendix D.1). We then establish a high-probability good event for the 1-bit estimators (Appendix D.2), verify the confidence width and retention of the optimal arm (Appendix D.3), and finally bound the cumulative regret (Appendix D.4).

Algorithm 1 1-Bit Phased Elimination with Uniform Arm Allocations

Input: Action set $\mathcal{A} \subset \mathbb{R}^d$, batch size B, confidence $\delta \in (0,1)$, variance σ^2, time horizon T.
Initialize epoch $h = 1$, active set $\mathcal{A}_1 = \mathcal{A}$.
Set $S_{\max} = 4d(\log\log d + 11)$ and $\widetilde{H} = \lceil \log_2 T \rceil$.
Set the number of Median-of-Means (MoM) blocks:

$$K = \left\lceil 8 \log\left(\frac{2S_{\max}\widetilde{H}}{\delta}\right)\right\rceil.$$

for epoch $h = 1, 2, \ldots$ **do**
  Find design $\rho_h \colon \mathcal{A}_h \to [0,1]$ satisfying (6).
  Set precision and confidence width:

$$\varepsilon_h = \frac{\sigma}{2^h\sqrt{B}}, \qquad W_h = \sqrt{2d}\cdot\varepsilon_h.$$

  for each arm $a \in \mathrm{supp}(\rho_h)$ **do**
    $[L_a, U_a] \leftarrow$ 1BitLocalize $\left(a,\ \delta_{\mathrm{loc}} = \dfrac{\delta}{2S_{\max}\widetilde{H}},\ \sigma' = \dfrac{\sigma}{\sqrt{B}}\right)$ (from Appendix B.1).
  end for
  Set the uniform arm budget u_h for all $a \in \mathrm{supp}(\rho_h)$:

$$u_h = B\left\lceil 16 C_V \cdot 4^h \cdot \log_2 T \right\rceil,$$

  where $C_V \geq 48$ is the absolute constant in the variance from Corollary 5.
  for each arm $a \in \mathrm{supp}(\rho_h)$ **do**
    for block $j = 1, \ldots, K$ **do**
      $\hat{Y}_{a,h}^{(j)} \leftarrow$ 1BitRefine $\left(a, [L_a, U_a],\ n_{\mathrm{q}} = \dfrac{u_h}{B}\right)$ (from Appendix B.2).
    end for
    Compute robust scalar estimate: $\hat{Y}_{a,h} = \mathrm{median}\{\hat{Y}_{a,h}^{(1)}, \ldots, \hat{Y}_{a,h}^{(K)}\}$.
  end for
  Compute the design-weighted global parameter estimate $\hat{\theta}_h \in \mathbb{R}^d$ according to (7) if $\mathrm{span}(\mathcal{A}_h) = \mathbb{R}^d$, or similarly with $\hat{\theta}_h \in \mathrm{span}(\mathcal{A}_h)$ otherwise (see Remark 6).
  for each arm $b \in \mathcal{A}_h$ **do**
    Compute $\hat{\mu}_{b,h} = b^\top \hat{\theta}_h$.
  end for
  Update active set: $\mathcal{A}_{h+1} = \{b \in \mathcal{A}_h : \max_{x \in \mathcal{A}_h} \hat{\mu}_{x,h} - \hat{\mu}_{b,h} \leq 2W_h\}$.
  Update sample counter; **break** when budget T is exhausted.
end for

Remark 8 (Finite-horizon execution). The algorithm runs with the prescribed allocations until T pulls have been made, meaning any subroutine may terminate early after the T-th pull. Only pulls that are actually made contribute to the regret. However, when upper bounding the regret of a stage, we may assume that every allocated pull is performed (even if the stage is actually interrupted), since additional pulls can only increase regret. For the probability analysis, "good event" conditions are

imposed only on completed stages, and if a specified conditioning point is not reached, we define the corresponding history to be the history at termination.

Throughout this section, let $\sigma' = \sigma/\sqrt{B}$ and $\varepsilon_h = 2^{-h}\sigma'$ as in Line 7 of Algorithm 1. We invoke Corollary 5 with the full-batch effective standard deviation $\sigma' = \sigma/\sqrt{B}$; each query corresponds to B raw arm pulls. Conditional on a fixed valid localization history, the guarantees from (20) are:

$$\mathrm{Var}(\hat{Y}) \le \frac{C_V \sigma'^2 \log_2 T}{n_{\mathrm{q}}}, \qquad |\mathrm{Bias}(\hat{Y})| \le \frac{3\sigma'}{T^2}, \tag{21}$$

where n_{q} denotes the number of *queries* (i.e., the number of raw samples divided by B) and $C_V \ge 48$ is the variance constant from Corollary 5. Crucially, the bias bound is independent of both n_{q} and the epoch index h.

D.1 Sample Complexity per Epoch

To facilitate the regret analysis, we separate the sample complexity of the localization and refinement stages in each epoch.

Localization cost. By Proposition 7, each call to 1BitLocalize with effective standard deviation $\sigma' = \sigma/\sqrt{B}$ and failure probability $\delta_{\mathrm{loc}} = \delta/(2S_{\max}\widetilde{H})$ uses at most

$$n_{\mathrm{loc}} = C_{\mathrm{loc}} \left(\log\left(\frac{1}{\sigma'}\right) + \log\left(\frac{1}{\delta_{\mathrm{loc}}}\right) \right) = C_{\mathrm{loc}} \left(\frac{1}{2}\log\left(\frac{B}{\sigma^2}\right) + \log\left(\frac{2S_{\max}\widetilde{H}}{\delta}\right) \right)$$

queries, where $C_{\mathrm{loc}} > 0$ is the absolute constant from Proposition 7. Since each query costs B raw arm pulls, and since $S_{\max} = O(d \log\log d)$, and $\widetilde{H} = \lceil \log_2 T \rceil$, the total localization cost over all support arms is bounded by

$$m_{\mathrm{loc},h} \le S_{\max} \cdot B \cdot n_{\mathrm{loc}} = O\left(dB \log\log d \cdot \log\left(\frac{dB \log T}{\delta}\right) \right), \tag{22}$$

where we have absorbed $\sigma = \Theta(1)$ into the $O(\cdot)$ notation.

Refinement cost. Because the uniform allocation u_h (see Line 11) is identical for every arm in the support, the total refinement cost $m_{\mathrm{ref},h}$ is at most $K \cdot |\operatorname{supp}(\rho_h)| \cdot u_h$ raw arm pulls. Using $|\operatorname{supp}(\rho_h)| \le S_{\max} = O(d \log\log d)$ and $u_h \le B(16C_V \cdot 4^h \log_2 T + 1)$, we have

$$m_{\mathrm{ref},h} \le K \cdot S_{\max} \cdot B\big(16C_V \cdot 4^h \log_2 T + 1\big) = O\big(KdB \log T \log\log d \cdot 4^h\big). \tag{23}$$

D.2 High-Probability Good Event

Let H be the last epoch of Algorithm 1 reached by time T, with $H = 0$ if none. Define the good event $\mathcal{E} = \mathcal{E}_{\mathrm{loc}} \cap \mathcal{E}_{\mathrm{ref}}$, where

$$\mathcal{E}_{\mathrm{loc}} := \bigcap_{h=1}^{H} \bigcap_{a \in \operatorname{supp}(\rho_h)} \left\{ \mu_a \in [L_a, U_a] \text{ and } U_a - L_a \le \tfrac{8\sigma}{\sqrt{B}} \right\},$$

$$\mathcal{E}_{\mathrm{ref}} := \bigcap_{h=1}^{H} \bigcap_{a \in \operatorname{supp}(\rho_h)} \left\{ |\hat{Y}_{a,h} - \mu_a| \le \varepsilon_h \right\}.$$

Lemma 11 (Good event probability)**.** $\mathbb{P}(\mathcal{E}) \ge 1 - \delta$.

Proof. We decompose the proof into multiple parts as follows.

Part 1: Deterministic bounds. We first state two deterministic bounds used throughout the proof.

1.1 Epoch count $H \le \widetilde{H}$. By Line 11 and $4^{\lceil \log_2 T \rceil} \ge T^2$, we have

$$u_{\widetilde{H}} = B\left\lceil 16C_V \cdot 4^{\widetilde{H}} \log_2 T \right\rceil \ge 16C_V B \cdot T^2 \log_2 T > T,$$

so epoch $\widetilde{H}$ cannot be completed within T pulls. The algorithm therefore cannot reach epoch $\widetilde{H}+1$, giving $H \le \widetilde{H} = \lceil \log_2 T \rceil$.

1.2 Core set size. By the near G-optimal design property (6), the core set satisfies $|\operatorname{supp}(\rho_h)| \le S_{\max}$ for every epoch $h \ge 1$.

Part 2: Localization event $\mathcal{E}_{\mathrm{loc}}$. The two bounds above imply that Algorithm 1 makes at most $S_{\max} \cdot \widetilde{H}$ total calls to 1BITLOCALIZE. By Proposition 7, each call uses fresh arm pulls and has failure probability at most $\delta_{\mathrm{loc}} = \delta/(2S_{\max}\widetilde{H})$ conditional on the history before that call. A union bound over all calls gives

$$\mathbb{P}(\mathcal{E}_{\mathrm{loc}}^c) \le \widetilde{H} \cdot S_{\max} \cdot \delta_{\mathrm{loc}} = \frac{\delta}{2}.$$

Part 3: Refinement event $\mathcal{E}_{\mathrm{ref}}$. We show $\mathbb{P}(\mathcal{E}_{\mathrm{ref}}^c \cap \mathcal{E}_{\mathrm{loc}}) \le \delta/2$.

Fix an epoch index $h \in \{1, \dots, \widetilde{H}\}$. Let $\mathcal{G}_h$ be the σ-algebra generated by the history immediately after localization in epoch h and before its refinement observations are drawn. Condition on a fixed realization of $\mathcal{G}_h$ in which localization is completed, all its intervals are valid, and enough pulls remain to complete the allocated refinement stage. This fixes $\mathcal{A}_h$, ρ_h, u_h, and the intervals $\{[L_a, U_a]\}_{a \in \operatorname{supp}(\rho_h)}$. The refinement observations and random thresholds are fresh. All expectations, biases, and variances below refer to this conditional law.

For any fixed $a \in \operatorname{supp}(\rho_h)$, since each block $j \in \{1, \dots, K\}$ uses fresh, independent arm pulls with the same fixed parameters, the K estimates $\hat{Y}_{a,h}^{(1)}, \dots, \hat{Y}_{a,h}^{(K)}$ are i.i.d., and so Lemma 5 applies to their median. We bound the bias and variance of a single block estimate.

- **Bias.** Since $n_{\mathrm{q}} = u_h/B \ge 16C_V \cdot 4^h \log_2 T \ge 48 \log_2 T$, Corollary 5 gives $|\mathrm{Bias}(\hat{Y}_{a,h}^{(1)})| \le 3\sigma'/T^2$. Since $h \le H \le \widetilde{H}$ and $2^{\widetilde{H}} = 2^{\lceil \log_2 T \rceil} \le 2T$, we have $2^h \le 2T$, so

$$\frac{3\sigma'}{T^2} \le \frac{6\sigma'}{T \cdot 2^h} = \frac{6\varepsilon_h}{T} \le \frac{\varepsilon_h}{2},$$

  where the last step holds for $T \ge 12$.

- **Variance.** Applying Corollary 5 with $n_{\mathrm{q}} = u_h/B$ and using the budget bound $u_h/B \ge 16C_V 4^h \log_2 T$ gives

$$\mathrm{Var}(\hat{Y}_{a,h}^{(1)}) \le \frac{C_V \sigma'^2 \log_2 T}{u_h/B} \le \frac{C_V \sigma'^2 \log_2 T}{16C_V \cdot 4^h \log_2 T} = \frac{\varepsilon_h^2}{16}.$$

  where the last equality uses $\varepsilon_h = \sigma'/2^h$.

Applying Lemma 5 to the K i.i.d. block estimates (i.e., $k = K$ therein) with variance at most $\varepsilon_h^2/16$ and a failure probability of $\delta/(2S_{\max}\widetilde{H})$, we obtain

$$\left|\hat{Y}_{a,h} - \mathbb{E}[\hat{Y}_{a,h}^{(1)}]\right| \le \sqrt{\frac{32 \cdot (\varepsilon_h^2/16)}{K} \log\left(\frac{2S_{\max}\widetilde{H}}{\delta}\right)} = \varepsilon_h \sqrt{\frac{2\log(2S_{\max}\widetilde{H}/\delta)}{K}} \le \frac{\varepsilon_h}{2},$$

where the last step uses $K \ge 8\log(2S_{\max}\widetilde{H}/\delta)$. Combined with the bias bound, the triangle inequality gives

$$|\hat{Y}_{a,h} - \mu_a| \le \underbrace{|\hat{Y}_{a,h} - \mathbb{E}[\hat{Y}_{a,h}^{(1)}]|}_{\le \varepsilon_h/2} + \underbrace{|\mathrm{Bias}(\hat{Y}_{a,h}^{(1)})|}_{\le \varepsilon_h/2} \le \varepsilon_h,$$

with probability at least $1 - \delta/(2S_{\max}\widetilde{H})$ under our fixed conditioning. Let $\mathcal{V}_h \in \mathcal{G}_h$ be the event that localization in epoch h is completed by time T and all its intervals are valid. Let F_h be the event that refinement in epoch h is completed by time T and $|\hat{Y}_{a,h} - \mu_a| > \varepsilon_h$ for some $a \in \operatorname{supp}(\rho_h)$. A union bound over $\operatorname{supp}(\rho_h)$, whose size is at most $S_{\max}$, gives $\mathbb{P}(F_h \mid \mathcal{G}_h) \le \delta/(2\widetilde{H})$ on $\mathcal{V}_h$.

Since $F_h \cap \mathcal{E}_{\text{loc}} \subseteq F_h \cap \mathcal{V}_h$ and $\mathcal{V}_h \in \mathcal{G}_h$, the tower property gives

$$\begin{aligned}\mathbb{P}(F_h \cap \mathcal{E}_{\text{loc}}) &\le \mathbb{P}(F_h \cap \mathcal{V}_h) \\ &= \mathbb{E}\left[\mathbb{E}\left[\mathbb{1}\{\mathcal{V}_h\} \cdot \mathbb{1}\{F_h\} \mid \mathcal{G}_h\right]\right] \\ &= \mathbb{E}\left[\mathbb{1}\{\mathcal{V}_h\} \cdot \mathbb{P}(F_h \mid \mathcal{G}_h)\right] \\ &\le \frac{\delta}{2\widetilde{H}}.\end{aligned} \tag{24}$$

Since $H \le \widetilde{H}$, a union bound over the fixed indices $h = 1, \dots, \widetilde{H}$ gives

$$\mathbb{P}(\mathcal{E}_{\text{ref}}^c \cap \mathcal{E}_{\text{loc}}) \le \sum_{h=1}^{\widetilde{H}} \frac{\delta}{2\widetilde{H}} = \frac{\delta}{2}\,.$$

Part 4: Conclusion. Combining this with the localization failure bound, we obtain

$$\mathbb{P}(\mathcal{E}^c) \le \mathbb{P}(\mathcal{E}_{\text{loc}}^c) + \mathbb{P}(\mathcal{E}_{\text{ref}}^c \cap \mathcal{E}_{\text{loc}}) \le \frac{\delta}{2} + \frac{\delta}{2} = \delta.$$

Thus $\mathbb{P}(\mathcal{E}) \ge 1 - \delta$, as desired. □

D.3 Confidence Width and Arm Elimination

Lemma 12 (Confidence width)**.** *Under event $\mathcal{E}$, for every epoch h whose refinement stage is completed and every $b \in \mathcal{A}_h$, we have $|\hat{\mu}_{b,h} - \mu_b| \le W_h = \sqrt{2d}\varepsilon_h$.*

Proof. Using (7) and the identity

$$\theta = G_h^{-1} \sum_{a \in \text{supp}(\rho_h)} \rho_h(a) a a^\top \theta \implies \mu_b = b^\top \theta = b^\top G_h^{-1} \sum_{a \in \text{supp}(\rho_h)} \rho_h(a) \cdot a \cdot \mu_a,$$

we have

$$|\hat{\mu}_{b,h} - \mu_b| = \left| b^\top G_h^{-1} \sum_a \rho_h(a) \cdot a \cdot (\hat{Y}_{a,h} - \mu_a) \right| \le \sum_a \left|\hat{Y}_{a,h} - \mu_a\right| \cdot \rho_h(a) \cdot \left|b^\top G_h^{-1} a\right|.$$

Under $\mathcal{E}_{\text{ref}}$, each $|\hat{Y}_{a,h} - \mu_a| \le \varepsilon_h$. Combining this with Corollary 4, we obtain $|\hat{\mu}_{b,h} - \mu_b| \le \varepsilon_h\sqrt{2d} = W_h$. □

Lemma 13 (Optimal arm never eliminated)**.** *Under $\mathcal{E}$, the optimal arm $a^* = \arg\max_{a \in \mathcal{A}} \mu_a$ satisfies $a^* \in \mathcal{A}_h$ for all epochs $h \ge 1$.*

Proof. We prove this by induction. The base case $\mathcal{A}_1 = \mathcal{A}$ is immediate. For the inductive step, suppose that $a^* \in \mathcal{A}_h$, and assume for contradiction that a^* is eliminated at epoch h. Then $\hat{\mu}_{b,h} - \hat{\mu}_{a^*,h} > 2W_h$ for some $b \in \mathcal{A}_h$. Lemma 12 gives $\mu_b - \mu_{a^*} \ge (\hat{\mu}_{b,h} - W_h) - (\hat{\mu}_{a^*,h} + W_h) > 0$, contradicting the optimality of a^*. □

Lemma 14 (Suboptimality gap for active arms)**.** *Under $\mathcal{E}$, for all epochs $h \ge 2$ and all active arms $b \in \mathcal{A}_h$, we have*

$$\Delta_b := \mu_{a^*} - \mu_b \le 4W_{h-1} = 8W_h.$$

Proof. Fix $h \ge 2$ and $b \in \mathcal{A}_h$. Since b survived the elimination step at the end of epoch $h-1$, and $a^* \in \mathcal{A}_{h-1}$ by Lemma 13, the elimination rule (see Line 22) gives

$$\hat{\mu}_{a^*,h-1} - \hat{\mu}_{b,h-1} \le \max_{x \in \mathcal{A}_{h-1}} \hat{\mu}_{x,h-1} - \hat{\mu}_{b,h-1} \le 2W_{h-1}.$$

Applying Lemma 12 to bound $\mu_{a^*} \le \hat{\mu}_{a^*,h-1} + W_{h-1}$ and $\mu_b \ge \hat{\mu}_{b,h-1} - W_{h-1}$, we obtain

$$\Delta_b = \mu_{a^*} - \mu_b \le (\hat{\mu}_{a^*,h-1} - \hat{\mu}_{b,h-1}) + 2W_{h-1} \le 4W_{h-1} = 8W_h,$$

where the last equality uses $W_{h-1} = 2W_h$ (see Line 7). □

D.4 Regret Bound

Proof of Theorem 3. We condition on the good event $\mathcal{E}$, which by Lemma 11 occurs with probability at least $1-\delta$. Throughout the remainder of this proof, all bounds are deterministic consequences of this event.

Let $\Delta_b = \mu_{a^*} - \mu_b$ denote the suboptimality gap of any active arm $b \in \mathcal{A}_h$. Because all mean rewards lie in $[-1,1]$, the gap is trivially bounded by $\Delta_b \le 2$. For epochs $h \ge 2$, Lemma 14 gives $\Delta_b \le \min\{8W_h, 2\}$.

We upper bound the total regret by splitting it into the regret incurred during the localization stages and the regret incurred during the refinement stages across all epochs. Let $H \le \lceil \log_2 T \rceil$ be the epoch at which the algorithm halts.

Localization Regret. Because the instantaneous regret is always at most 2, the total regret incurred during the localization phases across all H epochs is bounded via (22):

$$R_{\mathrm{loc}} \le \sum_{h=1}^{H} 2 \cdot m_{\mathrm{loc},h} = O\left(H \cdot dB \log\log d \log\left(\frac{dB\log T}{\delta}\right)\right) = \widetilde{O}\left(dB\log\left(\frac{1}{\delta}\right)\right), \tag{25}$$

where the $\widetilde{O}(\cdot)$ notation absorbs the polylog(d,T) factors.

Refinement Regret. For the first epoch ($h=1$), we bound the refinement gap trivially by 2:

$$R_{\mathrm{ref},1} \le 2 \cdot m_{\mathrm{ref},1} = O(KdB\log T\log\log d) = \widetilde{O}(K\cdot dB). \tag{26}$$

For any subsequent epoch $h \ge 2$, Lemma 14 gives $\Delta_b \le 8W_h$. It follows that the refinement regret for epoch $h \ge 2$ is bounded by

$$R_{\mathrm{ref},h} \le m_{\mathrm{ref},h} \cdot 8W_h = m_{\mathrm{ref},h} \cdot \frac{8\sqrt{2d}\sigma}{2^h\sqrt{B}} = O\left(2^{-h} \cdot m_{\mathrm{ref},h} \cdot \sqrt{\frac{d}{B}}\right). \tag{27}$$

Epochs with $m_{\mathrm{ref},h} = 0$ contribute no refinement regret. For the remaining epochs, taking the bound $m_{\mathrm{ref},h} \le O(KdB\log T\log\log d \cdot 4^h)$ established in (23) and rearranging yields

$$4^{-h} \le O\left(\frac{KdB\log T\log\log d}{m_{\mathrm{ref},h}}\right) \implies 2^{-h} \le O\left(\sqrt{\frac{KdB\log T\log\log d}{m_{\mathrm{ref},h}}}\right).$$

Substituting this bound into (27) allows the batch size B to cancel out, and we obtain

$$R_{\mathrm{ref},h} \le O\left(\sqrt{\frac{KdB\log T\log\log d}{m_{\mathrm{ref},h}}} \cdot m_{\mathrm{ref},h} \cdot \sqrt{\frac{d}{B}}\right) = O\left(d\sqrt{K\log T\log\log d} \cdot \sqrt{m_{\mathrm{ref},h}}\right).$$

Summing this over $h = 2,\dots,H$ and applying the Cauchy-Schwarz inequality and the trivial bound $\sum_{h=2}^{H} m_{\mathrm{ref},h} \le T$, we obtain

$$\sum_{h=2}^{H} \sqrt{m_{\mathrm{ref},h}} = \sum_{h=2}^{H} \left(\sqrt{m_{\mathrm{ref},h}} \cdot 1\right) \le \sqrt{\sum_{h=2}^{H} m_{\mathrm{ref},h}} \cdot \sqrt{\sum_{h=2}^{H} 1} \le \sqrt{T\cdot H}, \tag{28}$$

Since $H \le \lceil \log_2 T \rceil$, the total refinement regret for $h \ge 2$ is bounded by

$$\sum_{h=2}^{H} R_{\mathrm{ref},h} \le O\left(d\sqrt{K\log T\log\log d} \cdot \sqrt{T\log T}\right) = \widetilde{O}\left(d\sqrt{KT}\right). \tag{29}$$

Total regret. Combining the bounds from (25), (26), and (29), the cumulative regret satisfies

$$R_T \le R_{\mathrm{loc}} + R_{\mathrm{ref},1} + \sum_{h=2}^{H} R_{\mathrm{ref},h} = \widetilde{O}\left(dB\log\left(\frac{1}{\delta}\right) + K\cdot dB + d\sqrt{KT}\right). \tag{30}$$

Using $S_{\max} = O(d\log\log d)$ and $\widetilde{H} = \lceil \log_2 T \rceil$, we have

$$K = \left\lceil 8\log\left(\frac{2S_{\max}\widetilde{H}}{\delta}\right)\right\rceil = O\left(\log\left(\frac{d\log T}{\delta}\right)\right),$$

and so the middle term in (30) is absorbed by the first, giving the final expression

$$R_T = \widetilde{O}\left(dB \log\left(\frac{1}{\delta}\right) + d\sqrt{T \log\left(\frac{1}{\delta}\right)}\right). \tag{31}$$

This matches the bound stated in Theorem 3. □

E Proof of Theorem 4 ($|\mathcal{A}|$-Dependent Upper Bound)

In this section, we provide the pseudocode and detailed regret analysis for our second phased elimination algorithm, formally establishing the bounds stated in Theorem 4. The detailed pseudocode is presented in Algorithm 2. We again use the finite-horizon conventions in Remark 8, applying the completed-stage convention also to individual rounds of SAFEARM.

We split our analysis into two regimes: the small-B regime, where $B < 8K_\star d(\log\log d + 11) = \widetilde{O}(d\log(|\mathcal{A}|/\delta))$, and the large-$B$ regime, where B exceeds this threshold (see Line 2 of Algorithm 2 for the definition of $K_\star$). In the small-B regime, the pre-processing is safely bypassed and only Step 3 (phased elimination) is executed with the initial active set $\mathcal{A}_1 = \mathcal{A}$ (analyzed in Appendix E.1). In the large-B regime, Steps 1 and 2 (safe arm identification and a warm start epoch) are run first to produce a "warm-started" active set $\mathcal{A}_1 \subseteq \mathcal{A}$ before proceeding to Step 3 (analyzed in Appendix E.2). The main difference between the two regimes is the worst-case suboptimality gap available at the *first epoch* of Step 3: a trivial $O(1)$ bound in the small-B case versus the refined $O(\sigma\sqrt{d/B})$ guarantee from the warm start in the large-B case.

Remark 9 (Comparison with Algorithm 1). Algorithm 2 differs from Algorithm 1 in three interconnected ways, each stemming from the goal of capturing the tight $|\mathcal{A}|$-dependence required for smaller action sets (e.g., $|\mathcal{A}| = \mathrm{poly}(d)$).

- **Proportional vs. uniform allocation.** Algorithm 1 allocates a uniform query budget u_h to every core-set arm, so each scalar block estimate $\hat{Y}_{a,h}^{(j)}$ has variance $O(\varepsilon_h^2)$ conditional on a fixed valid localization history. Consequently, propagating these scalar error bounds through $\hat{\theta}_h$ via Corollary 4 gives the confidence width $W_h = \sqrt{2d}\,\varepsilon_h$. Algorithm 2 instead allocates $u_h(a) \propto \rho_h(a)$, meaning $\mathrm{Var}(\hat{Y}_{a,h}^{(j)}) \propto 1/\rho_h(a)$. These $\rho_h(a)$ factors perfectly cancel when computing $\mathrm{Var}(b^\top \hat{\theta}_h^{(j)})$ via Lemma 6, yielding the tighter confidence width $W_h = \varepsilon_h$. This variance cancellation is precisely what enables Algorithm 2 to match the optimal $\sigma\sqrt{dT\log|\mathcal{A}|}$ "standard linear bandit term" in the minimax lower bound.

- **Order of aggregation and larger K.** As noted in Section 4, the two algorithms differ in when the median-of-means aggregation is applied. Algorithm 1 applies it directly to the $|\mathrm{supp}(\rho_h)| = \widetilde{O}(d)$ scalar estimates $\hat{Y}_{a,h}^{(j)}$, meaning simultaneous concentration over the support set requires only $K = \widetilde{O}(\log(d/\delta))$ blocks. Algorithm 2 instead forms block-wise parameter estimates $\hat{\theta}_h^{(j)}$ first, and then evaluates the inner products $b^\top \hat{\theta}_h^{(j)}$ for all active arms $b \in \mathcal{A}_h$. We ensure simultaneous concentration over all active arms using a union bound over at most $|\mathcal{A}|$ arms. Accordingly, we use $K = \widetilde{O}(\log(|\mathcal{A}|/\delta))$ blocks, yielding the $\sqrt{\log(|\mathcal{A}|/\delta)}$ factor in the statistical term of Theorem 4.

- **Safe arm and warm start for large B.** In Algorithm 1, because $K = \widetilde{O}(\log(d/\delta))$, the first-epoch regret $\widetilde{O}(dBK) = \widetilde{O}(dB)$ matches the communication-based lower bound $\Omega(dB)$ (see Theorem 1) up to logarithmic factors, where we use $\Omega(B\min\{d, \log|\mathcal{A}|\}) = \Omega(dB)$ for the exponentially large action sets targeted by the algorithm. In Algorithm 2, however, the larger block count $K = \widetilde{O}(\log(|\mathcal{A}|/\delta))$ inflates the first-epoch cost to $\widetilde{O}(dB\log|\mathcal{A}|)$, which exceeds the communication-based lower bound $\Omega(B\log|\mathcal{A}|)$ by a factor of d and dominates the regret when B is large. The safe arm identification and warm-start steps (Algorithms 3 and 4) address this by reducing the first-epoch suboptimality gap from $O(1)$ down to $O(\sqrt{d/B})$ before the main loop begins, shrinking the first-epoch regret from $\widetilde{O}(dB\log|\mathcal{A}|)$ to $\widetilde{O}(d^{3/2}\sqrt{B}\log|\mathcal{A}|)$.

E.1 Small-B Regime: Step 3 as a Standalone Algorithm

In the small B regime where $B < 8K_\star d(\log\log d + 11) = \widetilde{O}(d\log(|\mathcal{A}|/\delta))$, Algorithm 2 skips Steps 1 and 2 and directly runs the phased elimination algorithm (Step 3) starting with $\mathcal{A}_1 = \mathcal{A}$. We first establish the shared technical machinery for the phased elimination (sample complexity, good event, confidence width, and arm elimination), which applies to both regimes. We then derive the small-B regret bound. The large-B analysis in Appendix E.2 reuses most of these lemmas unchanged, differing only in the first-epoch suboptimality gap.

Throughout this section, let $\sigma' = \frac{\sigma}{\sqrt{B}}$ and $\varepsilon_h = 2^{-h}\sigma'$ as defined in Line 13 of Algorithm 2. We invoke Corollary 5 with the full-batch effective standard deviation $\sigma' = \frac{\sigma}{\sqrt{B}}$; each query corresponds to B raw arm pulls. Conditional on a fixed valid localization history, the guarantees from (20) are:

$$\mathrm{Var}(\hat{Y}) \le \frac{C_V \sigma'^2 \log_2 T}{n_{\mathrm{q}}}, \qquad |\mathrm{Bias}(\hat{Y})| \le \frac{3\sigma'}{T^2}, \tag{32}$$

where n_{q} denotes the number of *queries* (i.e., the number of raw samples divided by B) and $C_V \ge 48$ is the variance constant from Corollary 5. The bias bound is independent of n_{q} and the epoch index h.

E.1.1 Sample Complexity per Epoch

Localization cost. By the same argument as in Appendix D.1, with failure probability $\delta_{\mathrm{loc}} = \delta/\big(6S_{\max}\widetilde{H}\big)$ in place of $\delta/\big(2S_{\max}\widetilde{H}\big)$, the total localization cost over all support arms satisfies

$$m_{\mathrm{loc},h} \;\le\; C_{\mathrm{loc}} \cdot S_{\max} \cdot B\left(\frac{1}{2}\log\frac{B}{\sigma^2} + \log\frac{6S_{\max}\widetilde{H}}{\delta}\right) \;=\; \widetilde{O}\left(dB\log\left(\frac{1}{\delta}\right)\right). \tag{33}$$

As in Appendix D, the $\log(1/\sigma)$ term is absorbed into the $\widetilde{O}(\cdot)$ notation since we assume $\sigma = \Theta(1)$.

Refinement cost. We bound the "true" refinement budget by summing the per-arm allocations $u_h(a)$ from Line 17 over the support and multiplying by K blocks. Using $\lceil x \rceil \le x + 1$ and $|\mathrm{supp}(\rho_h)| = \widetilde{O}(d)$, we bound the number of arm pulls used for refinement in an epoch by

$$\begin{aligned} m_{\mathrm{ref},h} &\le K \sum_{a\in\mathrm{supp}(\rho_h)} u_h(a) \\ &\le KB \sum_{a\in\mathrm{supp}(\rho_h)} \big(32C_V \cdot d \cdot \log_2 T \cdot 4^h \cdot \rho_h(a) + 48\log_2 T + 1\big) \\ &= O(dBK \cdot \log T \cdot \log\log d \cdot 4^h), \end{aligned} \tag{34}$$

The localization floor $\widetilde{O}(dB\log(1/\delta))$ is present at every epoch, while the refinement term $\widetilde{O}(dBK \cdot 4^h)$ grows geometrically and thus the largest h value is the dominant one.

E.1.2 High-Probability Good Event

Let H be the last epoch of Step 3 reached by time T, with $H = 0$ if none. Define the good event $\mathcal{E}_3 = \mathcal{E}_{3,\mathrm{loc}} \cap \mathcal{E}_{3,\mathrm{ref}}$, where

$$\mathcal{E}_{3,\mathrm{loc}} \coloneqq \bigcap_{h=1}^{H} \bigcap_{a\in\mathrm{supp}(\rho_h)} \left\{\mu_a \in [L_a, U_a] \;\text{ and }\; U_a - L_a \le \frac{8\sigma}{\sqrt{B}}\right\},$$

$$\mathcal{E}_{3,\mathrm{ref}} \coloneqq \bigcap_{h=1}^{H} \bigcap_{b\in\mathcal{A}_h} \{|\hat{\mu}_{b,h} - \mu_b| \le \varepsilon_h\}.$$

Lemma 15 (Good event probability for Step 3)**.** $\mathbb{P}(\mathcal{E}_3) \ge 1 - \delta/3$. *The same bound holds conditional on the history before Step 3.*

Algorithm 2 1-bit Phased Elimination with Safe Arm and Warm Start

Input: Action set $\mathcal{A} \subset \mathbb{R}^d$, batch size B, confidence $\delta \in (0,1)$, variance σ^2, time horizon T.
Set $K_\star = \left\lceil 8 \log\left(\frac{3|\mathcal{A}| \cdot \lceil \log_2 |\mathcal{A}| \rceil}{\delta}\right)\right\rceil$.
if $B \geq 8K_\star d(\log\log d + 11)$ **then**
  Compute $a_{\text{safe}} \leftarrow \textsc{SafeArm}(\mathcal{A}, B, \delta)$ (Algorithm 3 in Appendix E.3.1).
  Set $\mathcal{A}_1 \leftarrow \textsc{WarmStart}(\mathcal{A}, a_{\text{safe}}, B, \delta)$ (Algorithm 4 in Appendix E.3.2).
else
  Set $\mathcal{A}_1 \leftarrow \mathcal{A}$
end if
Set $S_{\max} = 4d(\log\log d + 11)$ and $\widetilde{H} = \lceil \log_2 T \rceil$.
Set the number of Median-of-Means (MoM) blocks:

$$K = \left\lceil 8 \log\left(\frac{6|\mathcal{A}|\widetilde{H}}{\delta}\right)\right\rceil.$$

for epoch $h = 1, 2, \ldots$ **do**
  Find design $\rho_h \colon \mathcal{A}_h \to [0,1]$ satisfying (6).
  Set precision and confidence width:

$$\varepsilon_h = \frac{\sigma}{2^h\sqrt{B}}, \qquad W_h = \varepsilon_h,$$

  where $C_V \geq 48$ is the variance constant from Corollary 5.
  for each arm $a \in \operatorname{supp}(\rho_h)$ **do**
    $[L_a, U_a] \leftarrow \textsc{1BitLocalize}\left(a,\ \delta_{\text{loc}} = \frac{\delta}{6S_{\max}\widetilde{H}},\ \sigma' = \frac{\sigma}{\sqrt{B}}\right)$ (from Appendix B.1).
  end for
  Set per-arm sample budget for each block:

$$u_h(a) = B \max\left\{\lceil 48 \log_2 T \rceil,\ \left\lceil 32 C_V \cdot d \log_2 T \cdot 4^h \cdot \rho_h(a) \right\rceil\right\}.$$

  for block $j = 1, \ldots, K$ **do**
    Compute $\hat{Y}_{a,h}^{(j)} \leftarrow \textsc{1BitRefine}\,(a, [L_a, U_a],\ n_{\text{q}} = u_h(a)/B)$ (from Appendix B.2) for each $a \in \operatorname{supp}(\rho_h)$.
    Compute the design-weighted parameter estimate $\hat{\theta}_h^{(j)} \in \mathbb{R}^d$ according to (8) if $\operatorname{span}(\mathcal{A}_h) = \mathbb{R}^d$, or similarly with $\hat{\theta}_h^{(j)} \in \operatorname{span}(\mathcal{A}_h)$ otherwise (see Remark 6).
  end for
  for each arm $b \in \mathcal{A}_h$ **do**
    $\hat{\mu}_{b,h} = \operatorname{median}\left\{b^\top \hat{\theta}_h^{(1)}, \ldots, b^\top \hat{\theta}_h^{(K)}\right\}$.
  end for
  Update active set: $\mathcal{A}_{h+1} = \{a \in \mathcal{A}_h : \max_{b \in \mathcal{A}_h} \hat{\mu}_{b,h} - \hat{\mu}_{a,h} \leq 2W_h\}$.
  Update sample counter; **break** when budget T is exhausted.
end for

Proof. The proof follows the same structure as Lemma 11; we detail only the steps that differ, primarily the shift from bounding individual scalar estimates $\hat{Y}_{a,h}$ to bounding the aggregated estimates $b^\top \hat{\theta}_h^{(j)}$, and union-bounding over the active sets $\mathcal{A}_h$ rather than the core sets $\operatorname{supp}(\rho_h)$.

Part 1: Deterministic bounds. Parts 1.1 and 1.2 of Lemma 11 apply similarly. By Lines 13 and 17 of Algorithm 2, if epoch $\widetilde{H}$ is reached, its full refinement allocation satisfies

$$K \sum_{a \in \operatorname{supp}(\rho_{\widetilde{H}})} u_{\widetilde{H}}(a) \geq 32 C_V dBK \log_2 T \cdot 4^{\widetilde{H}} \geq 32 C_V dBK \log_2 T \cdot T^2 > T,$$

where the first inequality uses $u_{\widetilde{H}}(a) \geq 32 C_V dB \log_2 T \cdot 4^{\widetilde{H}} \rho_{\widetilde{H}}(a)$ from Line 17 and $\sum_a \rho_{\widetilde{H}}(a) = 1$. Thus epoch $\widetilde{H}$ cannot be completed within T pulls, giving $H \leq \widetilde{H}$ deterministically. The core set

bound $|\operatorname{supp}(\rho_h)| \le S_{\max}$ holds by (6). Finally, since the active set is obtained by successive elimination, we have $\mathcal{A}_h \subseteq \mathcal{A}$, and hence $|\mathcal{A}_h| \le |\mathcal{A}|$ for every epoch h.

Part 2: Localization event $\mathcal{E}_{3,\mathrm{loc}}$. This is similar to Part 2 of Lemma 11, with failure probability $\delta_{\mathrm{loc}} = \delta/(6S_{\max}\widetilde{H})$. Taking the union bound gives $\mathbb{P}(\mathcal{E}_{3,\mathrm{loc}}^c) \le \delta/6$.

Part 3: Refinement event $\mathcal{E}_{3,\mathrm{ref}}$. We show that $\mathbb{P}(\mathcal{E}_{3,\mathrm{ref}}^c \cap \mathcal{E}_{3,\mathrm{loc}}) \le \delta/6$.

Fix an epoch index $h \in \{1, \dots, \widetilde{H}\}$. As in Part 3 of Lemma 11, condition on a fixed post-localization history $\mathcal{G}_h$ with valid intervals and enough remaining pulls for the allocated refinement stage. The active set, design, allocations, and intervals are fixed, while the refinement observations and random thresholds are fresh. All expectations, biases, and variances below refer to this conditional law. For any fixed $b \in \mathcal{A}_h$, the K block estimates $b^\top \hat{\theta}_h^{(1)}, \dots, b^\top \hat{\theta}_h^{(K)}$ are i.i.d. by the same fresh-samples argument. Before bounding their bias and variance, we note that each invocation of 1BITREFINE uses a sample budget $n_{\mathrm{q}}(a) = u_h(a)/B$ that satisfies two simultaneous lower bounds via Line 17: $n_{\mathrm{q}}(a) \ge 48 \log_2 T$ ensuring Corollary 5 applies; and $n_{\mathrm{q}}(a) \ge 32 C_V \cdot d \cdot \log_2 T \cdot 4^h \cdot \rho_h(a)$, which yields

$$\frac{1}{n_{\mathrm{q}}(a)} \le \frac{1}{32 C_V \cdot d \cdot \log_2 T \cdot 4^h \cdot \rho_h(a)} \tag{35}$$

for the upcoming variance bound. We then compute the following:

- *Bias.* By writing $\mu_b = b^\top G_h^{-1} G_h \theta$ and using the definitions of G_h and $\hat{\theta}_h^{(1)}$ from (8), we can decompose the bias as

$$\mathbb{E}\left[b^\top \hat{\theta}_h^{(1)}\right] - \mu_b = \sum_{a \in \operatorname{supp}(\rho_h)} \rho_h(a) \cdot (b^\top G_h^{-1} a) \cdot \left(\mathbb{E}[\hat{Y}_{a,h}^{(1)}] - \mu_a\right).$$

  Applying the triangle inequality together with Corollaries 5 and 4, we obtain

$$\left|\mathbb{E}[b^\top \hat{\theta}_h^{(1)}] - \mu_b\right| \le \frac{3\sigma'}{T^2} \sum_a \rho_h(a) |b^\top G_h^{-1} a| \le \frac{3\sqrt{2d}\,\sigma'}{T^2} = \frac{3\sqrt{2d} \cdot 2^h \varepsilon_h}{T^2} \le \frac{6\sqrt{2d}\varepsilon_h}{T} \le \frac{\varepsilon_h}{2},$$

  where the last two steps use $2^h \le 2T$ and the allocated refinement stage fits within the horizon:

$$T \ge K \sum_{a \in \operatorname{supp}(\rho_h)} u_h(a) \ge 128 C_V d B K \log_2 T \ge 128 d \ge 12\sqrt{2d}.$$

- *Variance.* By the definition of $\hat{\theta}_h^{(1)}$ from (8) and the independence of the 1-bit estimates, we can write the variance as

$$\operatorname{Var}(b^\top \hat{\theta}_h^{(1)}) = \sum_a \rho_h(a)^2 \cdot (b^\top G_h^{-1} a)^2 \cdot \operatorname{Var}\left(\hat{Y}_{a,h}^{(1)}\right).$$

  Applying Corollary 5 with bound (35) and Lemma 6,

$$\begin{aligned}
\operatorname{Var}(b^\top \hat{\theta}_h^{(1)}) &\le \sum_a \rho_h(a)^2 \cdot (b^\top G_h^{-1} a)^2 \cdot \frac{C_V \sigma'^2 \log_2 T}{32 C_V \cdot d \cdot \log_2 T \cdot 4^h \cdot \rho_h(a)} \\
&\le \frac{\sigma'^2}{32 d \cdot 4^h} \sum_a \rho_h(a) \cdot (b^\top G_h^{-1} a)^2 \\
&\le \frac{\sigma'^2}{16 \cdot 4^h} \\
&= \frac{\varepsilon_h^2}{16},
\end{aligned}$$

  where the last step substitutes $\varepsilon_h = \sigma'/2^h$.

Applying Lemma 5 with failure probability $\delta/(6|\mathcal{A}|\widetilde{H})$ and combining with the bias bound via the triangle inequality gives $|\hat{\mu}_{b,h} - \mu_b| \le \varepsilon_h$ with probability at least $1 - \delta/(6|\mathcal{A}|\widetilde{H})$ under our fixed

conditioning. Use $\mathcal{G}_h$ and $\mathcal{V}_h$ with the same conventions as in Part 3 of Lemma 11, and let F_h be the event that refinement in epoch h is completed by time T and $|\hat{\mu}_{b,h} - \mu_b| > \varepsilon_h$ for some $b \in \mathcal{A}_h$. A union bound over $\mathcal{A}_h$, using $|\mathcal{A}_h| \le |\mathcal{A}|$, gives $\mathbb{P}(F_h \mid \mathcal{G}_h) \le \delta/(6\widetilde{H})$ on $\mathcal{V}_h$. Applying the averaging argument in (24) with $\mathcal{E}_{3,\mathrm{loc}}$ in place of $\mathcal{E}_{\mathrm{loc}}$, then taking a union bound over the fixed epoch indices, gives

$$\mathbb{P}(\mathcal{E}_{3,\mathrm{ref}}^c \cap \mathcal{E}_{3,\mathrm{loc}}) \le \sum_{h=1}^{\widetilde{H}} \frac{\delta}{6\widetilde{H}} = \frac{\delta}{6}.$$

Part 4: Conclusion. Combining this with the localization failure bound gives

$$\mathbb{P}(\mathcal{E}_3^c) \le \mathbb{P}(\mathcal{E}_{3,\mathrm{loc}}^c) + \mathbb{P}(\mathcal{E}_{3,\mathrm{ref}}^c \cap \mathcal{E}_{3,\mathrm{loc}}) \le \frac{\delta}{6} + \frac{\delta}{6} = \frac{\delta}{3}.$$

Hence, $\mathbb{P}(\mathcal{E}_3) \ge 1 - \delta/3$. Let $\mathcal{F}_3$ be the σ-algebra generated by the history before Step 3. The same proof applies conditional on $\mathcal{F}_3$, giving $\mathbb{P}(\mathcal{E}_3 \mid \mathcal{F}_3) \ge 1 - \delta/3$. □

E.1.3 Confidence Width and Arm Elimination

Based on $\mathcal{E}_{3,\mathrm{ref}} \supseteq \mathcal{E}_3$ in the good event, we immediately have the following guarantee on the confidence width.

Lemma 16 (Confidence width for Step 3)**.** *Under $\mathcal{E}_3$, for every epoch h whose refinement stage is completed and every active arm $b \in \mathcal{A}_h$, the estimation error satisfies $|\hat{\mu}_{b,h} - \mu_b| \le \varepsilon_h = W_h$.*

Lemma 17 (Optimal arm never eliminated)**.** *Under $\mathcal{E}_3$, the optimal arm $a^* = \arg\max_{a \in \mathcal{A}} \mu_a$ satisfies $a^* \in \mathcal{A}_h$ for all epochs $h \ge 1$.*

Proof. Identical to Lemma 13, replacing $\mathcal{E}$ with $\mathcal{E}_3$ and Lemma 12 with Lemma 16. □

Lemma 18 (Suboptimality gap for active arms, Step 3)**.** *Under $\mathcal{E}_3$, for all epochs $h \ge 2$ and all active arms $b \in \mathcal{A}_h$,*

$$\Delta_b := \mu_{a^*} - \mu_b \le 4W_{h-1} = 8W_h.$$

Proof. Identical to Lemma 14, replacing $\mathcal{E}$ with $\mathcal{E}_3$, Lemma 13 with Lemma 17, and Lemma 12 with Lemma 16. □

E.1.4 Regret Bound for the Small-B Regime

Theorem 19 (Regret of Step 3 without Warm Start)**.** *Under $\mathcal{E}_3$, the cumulative regret incurred by the standalone phased elimination algorithm (i.e., Algorithm 2 without warm start) satisfies*

$$R_T = \widetilde{O}\left(dB \log\left(\frac{|\mathcal{A}|}{\delta}\right) + \sqrt{dT \log\left(\frac{|\mathcal{A}|}{\delta}\right)} \right).$$

Proof. We upper bound the total regret by splitting it into localization and refinement regret. Let $H \le \widetilde{H} = \lceil \log_2 T \rceil$ be the final epoch at which the algorithm halts.

Localization Regret. Since the instantaneous regret is at most 2 throughout, the total localization regret over all H epochs is bounded via (33):

$$R_{\mathrm{loc}} \le \sum_{h=1}^{H} 2m_{\mathrm{loc},h} = O\left(H \cdot dB \cdot \log\log d \cdot \log\left(\frac{dB \log T}{\delta}\right)\right) = \widetilde{O}\left(dB \log\left(\frac{1}{\delta}\right)\right), \quad (36)$$

where the $\widetilde{O}(\cdot)$ notation absorbs the polylog(d, T) factors.

Refinement Regret. For the first epoch ($h = 1$), the trivial gap bound $\Delta_b \le 2$ and the upper bound (34) give

$$R_{\mathrm{ref},1} \le 2 \cdot m_{\mathrm{ref},1} = O(dBK \cdot \log T \cdot \log\log d) = \widetilde{O}(dBK). \quad (37)$$

For epochs $h \geq 2$, Lemma 18 gives $\Delta_b \leq 8W_h = 8\varepsilon_h = 8\sigma/(2^h\sqrt{B})$ for all active arms $b \in \mathcal{A}_h$. The refinement regret for epoch h is therefore

$$R_{\mathrm{ref},h} \leq m_{\mathrm{ref},h} \cdot 8W_h = m_{\mathrm{ref},h} \cdot \frac{8\sigma}{2^h\sqrt{B}}.$$

Epochs with $m_{\mathrm{ref},h} = 0$ contribute no refinement regret. For the remaining epochs, taking the bound $m_{\mathrm{ref},h} = O(dBK \cdot \log T \cdot \log\log d \cdot 4^h)$ established in (34) and rearranging yields

$$4^{-h} = O\left(\frac{dBK \cdot \log T \cdot \log\log d}{m_{\mathrm{ref},h}}\right) \implies 2^{-h} = O\left(\sqrt{\frac{dBK \cdot \log T \cdot \log\log d}{m_{\mathrm{ref},h}}}\right).$$

Substituting and using $\sigma = \Theta(1)$ yields

$$R_{\mathrm{ref},h} = O\left(\sqrt{dK \cdot \log T \cdot \log\log d \cdot m_{\mathrm{ref},h}}\right).$$

Summing over $h = 2, \ldots, H$ and applying Cauchy-Schwarz inequality as in (28), we obtain

$$\sum_{h=2}^{H} R_{\mathrm{ref},h} = O\left(\sqrt{dK \log T \log\log d} \cdot \sum_{h=2}^{H} \sqrt{m_{\mathrm{ref},h}}\right) = \widetilde{O}\left(\sqrt{dKT}\right). \tag{38}$$

Total Regret. Combining (36), (37), and (38),

$$R_T \leq R_{\mathrm{loc}} + R_{\mathrm{ref},1} + \sum_{h=2}^{H} R_{\mathrm{ref},h} = \widetilde{O}\left(dB\log\left(\frac{1}{\delta}\right) + dBK + \sqrt{KdT}\right).$$

Substituting $K = \lceil 8\log(6|\mathcal{A}|\widetilde{H}/\delta)\rceil = O(\log(|\mathcal{A}|\log T/\delta))$ yields

$$R_T = \widetilde{O}\left(dB\log\left(\frac{|\mathcal{A}|}{\delta}\right) + \sqrt{dT\log\left(\frac{|\mathcal{A}|}{\delta}\right)}\right) \tag{39}$$

as stated in Theorem 19. □

In the small-B regime, $B = \widetilde{O}(d\log(|\mathcal{A}|/\delta))$ implies $dB = \widetilde{O}(d^{3/2}\sqrt{B\log(|\mathcal{A}|/\delta)})$, and so the first term in (39) scales as $d^{3/2}\sqrt{B}\log^{\frac{3}{2}}(|\mathcal{A}|/\delta)$ up to logarithmic factors, meaning the regret is upper bounded by

$$R_T = \widetilde{O}\left(d^{3/2}\sqrt{B}\log^{\frac{3}{2}}\left(\frac{|\mathcal{A}|}{\delta}\right) + \sqrt{dT\log\left(\frac{|\mathcal{A}|}{\delta}\right)}\right).$$

E.2 Regret Bounds for Large B

We first explain why Theorem 19 is insufficient for large B. The first-epoch regret $\widetilde{O}(dBK)$ in that theorem arises from the trivial gap bound $\Delta_b \leq 2$: without prior information, no tighter bound is available at epoch 1. When $B \gg d$, this is wasteful because the dBK term can dominate both the communication bottleneck $\Omega(B\log|\mathcal{A}|)$ and the standard linear bandit term $\Omega(\sqrt{dT\log|\mathcal{A}|})$. A key observation towards improving this is that when $B \gg d$, a single batch is large enough to run a full experimental design *within the batch*.

Specifically, the first two steps of Algorithm 2 address this by prepending two pre-processing stages before the main phased elimination loop. Step 1 identifies a *safe arm* a_{safe} with suboptimality gap $\widetilde{O}(\log^{3/2}(|\mathcal{A}|/\delta)\sqrt{d/B})$. Step 2 uses a_{safe} to run a *warm-start epoch* that filters the initial active set down to $\mathcal{A}_1 \subseteq \mathcal{A}$ with $\Delta_b = O(\sqrt{d/B})$ for all $b \in \mathcal{A}_1$. With this tighter first-epoch gap bound, the first-epoch regret shrinks from $\widetilde{O}(dBK)$ to an improved term of $\widetilde{O}(d^{3/2}\sqrt{BK})$.

We now present the pseudocode for each pre-processing stage, followed by its regret analysis. Throughout, we assume $B \geq 8K_\star d(\log\log d + 11)$, where $K_\star = \lceil 8\log(3|\mathcal{A}|\lceil\log_2|\mathcal{A}|\rceil/\delta)\rceil$.

Algorithm 3 SAFEARM$(\mathcal{A}, B, \delta)$: Safe Arm Identification

Input: Action set $\mathcal{A} \subset \mathbb{R}^d$, batch size B, confidence δ.
Initialize: Active set $\mathcal{A}_1 = \mathcal{A}$.
Set number of Median-of-Means (MoM) blocks $K_\star = \left\lceil 8 \log\left(\frac{3|\mathcal{A}| \cdot \lceil \log_2 |\mathcal{A}| \rceil}{\delta}\right)\right\rceil$.
for $h = 1$ **to** $\lceil \log_2 |\mathcal{A}| \rceil$ **do**
  if $|\mathcal{A}_h| = 1$ **then**
    return the unique arm in $\mathcal{A}_h$ as a_{safe}.
  end if
  Arbitrarily split $\mathcal{A}_h$ into two equal groups $\mathcal{G}_A, \mathcal{G}_B$ (up to rounding).
  Compute design $\rho_h \colon \mathcal{A}_h \to [0,1]$ satisfying (6).
  Assign pulls $u_h(a) = \lceil B\rho_h(a)/(2K_\star) \rceil$ for each $a \in \operatorname{supp}(\rho_h)$.

Agent execution:
  for block $j = 1, \ldots, K_\star$ **do**
    Pull arm a a total of $u_h(a)$ times and compute sample mean $\bar{Y}_{a,h}^{(j)}$ of rewards for each arm $a \in \operatorname{supp}(\rho_h)$.
    Compute $\hat{\theta}_h^{(j)} \in \mathbb{R}^d$ according to (8) if $\operatorname{span}(\mathcal{A}_h) = \mathbb{R}^d$, or similarly with $\hat{\theta}_h^{(j)} \in \operatorname{span}(\mathcal{A}_h)$ otherwise (see Remark 6).
  end for
  Compute reward estimate $\hat{\mu}_{b,h} = \operatorname{median}\{b^\top \hat{\theta}_h^{(1)}, \ldots, b^\top \hat{\theta}_h^{(K_\star)}\}$ for each $b \in \mathcal{A}_h$.
  Transmit bit $Z_h = \mathbb{1}\{\max_{i \in \mathcal{G}_A} \hat{\mu}_{i,h} \geq \max_{j \in \mathcal{G}_B} \hat{\mu}_{j,h}\}$ to learner.

Learner performs elimination:
  if $Z_h = 1$ **then** $\mathcal{A}_{h+1} \leftarrow \mathcal{G}_A$ **else** $\mathcal{A}_{h+1} \leftarrow \mathcal{G}_B$.
end for
Output: The single arm remaining in $\mathcal{A}_{\lceil \log_2 |\mathcal{A}| \rceil + 1}$ as a_{safe}.

E.2.1 Step 1: Safe Arm Identification

Algorithm 3 runs at most $\lceil \log_2 |\mathcal{A}| \rceil$ rounds of tournament-style bisection. In each round, the active set is split in two, both halves are evaluated via a G-optimal design using the unquantized sample means available within a single batch, and the half with the higher estimated maximum is retained. After at most $\lceil \log_2 |\mathcal{A}| \rceil$ rounds, a single arm remains.

Regret of Step 1. Algorithm 3 uses at most B arm pulls in total per round over at most $\lceil \log_2 |\mathcal{A}| \rceil$ rounds, so the total number of arm pulls is at most $B\lceil \log_2 |\mathcal{A}| \rceil$. Since the instantaneous regret is at most 2, the regret contribution R_{safe} in this part satisfies

$$R_{\text{safe}} \leq 2B\lceil \log_2 |\mathcal{A}| \rceil = O(B \log |\mathcal{A}|).$$

Guarantee of Step 1. The following lemma, proved in Appendix E.3.1, bounds the suboptimality of the returned arm.

Lemma 20. *(Safe Arm Guarantees) Denote the output of the* SAFEARM *algorithm as* $a_{\text{safe}} \in \mathcal{A}$. *Then for any* $B \geq 8K_\star d(\log\log d + 11)$, *the event*

$$\mathcal{E}_1 := \left\{ \Delta_{a_{\text{safe}}} \leq 16\sigma \lceil \log_2 |\mathcal{A}| \rceil \sqrt{\frac{2d}{B} \log\left(\frac{3|\mathcal{A}| \cdot \lceil \log_2 |\mathcal{A}| \rceil}{\delta}\right)} \right\} \tag{40}$$

holds with probability at least $1 - \delta/3$.

E.2.2 Step 2: Warm Start

Algorithm 4 runs a single epoch of phased elimination with a key twist: in every batch, a_{safe} fills all slots not needed by a core-set arm. This means core-set arms are pulled only $u_0(a) = \lceil B\rho_0(a) \rceil \leq B$ times per batch, while a_{safe} absorbs the remaining $B - u_0(a)$ pulls. The effective standard deviation for estimating μ_a is therefore $\sigma/\sqrt{u_0(a)}$ rather than $\sigma/\sqrt{B}$, which is the key difference from the main

Algorithm 4 WARMSTART$(\mathcal{A}, a_{\text{safe}}, B, \delta)$: Warm Start Epoch

Input: Finite action set $\mathcal{A} \subset \mathbb{R}^d$, safe arm $a_{\text{safe}} \in \mathcal{A}$, batch size B, confidence δ.
Set $K_0 = \lceil 8 \log(6|\mathcal{A}|/\delta) \rceil$ and $S_{\max} = 4d(\log\log d + 11)$.
Compute design $\rho_0 \colon \mathcal{A} \to [0,1]$ satisfying (6).
for each $a \in \operatorname{supp}(\rho_0)$ **do**
  Set $u_0(a) = \lceil B\rho_0(a) \rceil$.
  $[L_a, U_a] \leftarrow$ 1BITLOCALIZE$\Big(a,\ \delta_{\text{loc}} = \dfrac{\delta}{6S_{\max}},\ \sigma' = \dfrac{\sigma}{\sqrt{u_0(a)}}\Big)$ (from Appendix B.1), with a_{safe} filling the remaining $B - u_0(a)$ slots of each batch.
end for
for block $j = 1, \ldots, K_0$ **do**
  $\hat{Y}_{a,0}^{(j)} \leftarrow$ 1BITREFINE$(a, [L_a, U_a], \lceil C_V \log_2 T \rceil)$ (from Appendix B.2) for each $a \in \operatorname{supp}(\rho_0)$, with a_{safe} filling remaining slots.
  Compute $\hat{\theta}_0^{(j)}$ via (8) if $\operatorname{span}(\mathcal{A}) = \mathbb{R}^d$, or $\hat{\theta}_0^{(j)} \in \operatorname{span}(\mathcal{A})$ otherwise (see Remark 6).
end for
for each $b \in \mathcal{A}$ **do**
  $\hat{\mu}_{b,0} = \operatorname{median}\{b^\top \hat{\theta}_0^{(1)}, \ldots, b^\top \hat{\theta}_0^{(K_0)}\}$.
end for
Output: $\mathcal{A}_1 = \Big\{a \in \mathcal{A} : \max_{b \in \mathcal{A}} \hat{\mu}_{b,0} - \hat{\mu}_{a,0} \le 8\sigma\sqrt{2d/B}\Big\}$.

phased elimination loop. Arms whose estimated gap exceeds $8\sigma\sqrt{2d/B}$ are eliminated, producing the warm-started active set $\mathcal{A}_1$.

Regret of Step 2. The warm-start regret has three components: localization, refinement, and safe arm pulls. Since $B \ge 8K_\star d(\log\log d + 11) \ge 2S_{\max} \ge |\operatorname{supp}(\rho_0)|$ by assumption and (6), and $\lceil x \rceil \le x + 1$, we have

$$\sum_{a \in \operatorname{supp}(\rho_0)} u_0(a) \le \sum_{a \in \operatorname{supp}(\rho_0)} (B\rho_0(a) + 1) = B + |\operatorname{supp}(\rho_0)| \le 2B.$$

Localization. By Proposition 7 with $\sigma' = \sigma/\sqrt{u_0(a)}$ and failure probability $\delta_{\text{loc}} = \delta/(6S_{\max})$, each invocation of 1BITLOCALIZE$(a, \delta_{\text{loc}}, \sigma')$ uses at most

$$n_{\text{loc},0}(a) = C_{\text{loc}} \left(\frac{1}{2} \log\left(\frac{u_0(a)}{\sigma^2} \right) + \log\left(\frac{6S_{\max}}{\delta} \right) \right) = O\left(\log\left(\frac{dB}{\delta} \right) \right)$$

queries, where we use $u_0(a) \le B$, and $\sigma = \Theta(1)$, and $S_{\max} = O(d \log\log d) = \widetilde{O}(d)$. Since each query costs $u_0(a)$ arm pulls and the regret of each pull is at most 2, summing the regret over $\operatorname{supp}(\rho_0)$ yields a contribution $R_{\text{ws,loc}}$ to the regret satisfying

$$R_{\text{ws,loc}} \le 2 \sum_{a \in \operatorname{supp}(\rho_0)} u_0(a) \cdot n_{\text{loc},0}(a) = O\left(B \log\left(\frac{dB}{\delta} \right) \right).$$

Refinement. Each block uses $\lceil C_V \log_2 T \rceil$ queries per arm, each costing $u_0(a)$ pulls and each pull incurs regret at most 2. Summing over $K_0 = O(\log(|\mathcal{A}|/\delta))$ blocks and $\operatorname{supp}(\rho_0)$ yields a contribution $R_{\text{ws,ref}}$ to the regret satisfying

$$R_{\text{ws,ref}} \le 2K_0 \sum_{a \in \operatorname{supp}(\rho_0)} u_0(a) \cdot \lceil C_V \log_2 T \rceil = O\left(B \log\left(\frac{|\mathcal{A}|}{\delta} \right) \log T \right).$$

Safe arm pulls. From the above discussion, the total number of batches consumed by localization and refinement is at most

$$\sum_{a \in \operatorname{supp}(\rho_0)} (n_{\text{loc},0}(a) + K_0 \cdot \lceil C_V \log_2 T \rceil) = O\left(S_{\max} \cdot \log\left(\frac{dB|\mathcal{A}|}{\delta} \right) \cdot \log(T) \right).$$

In each batch, a_{safe} is pulled at most B times. Conditioned on $\mathcal{E}_1$, Lemma 20 gives $\Delta_{a_{\text{safe}}} = \widetilde{O}\Big(\log|\mathcal{A}|\sqrt{\frac{d}{B}\log\frac{|\mathcal{A}|}{\delta}}\Big)$, so we get a contribution $R_{\text{ws,safe}}$ to the regret satisfying

$$R_{\text{ws,safe}} \le \Delta_{a_{\text{safe}}} \cdot O\left(B \cdot S_{\max} \cdot \log\left(\frac{dB|\mathcal{A}|}{\delta}\right) \cdot \log(T)\right) = \widetilde{O}\left(d^{3/2}\sqrt{B} \cdot \log^{5/2}\left(\frac{|\mathcal{A}|}{\delta}\right)\right).$$

Combining all three components, we obtain a total contribution R_{ws} satisfying

$$R_{\text{ws}} = R_{\text{ws,loc}} + R_{\text{ws,ref}} + R_{\text{ws,safe}} = \widetilde{O}\left(B\log\left(\frac{|\mathcal{A}|}{\delta}\right) + d^{3/2}\sqrt{B}\log^{5/2}\left(\frac{|\mathcal{A}|}{\delta}\right)\right). \tag{41}$$

Guarantee of Step 2. The following lemma, proved in Appendix E.3.2, bounds the suboptimality of every arm in $\mathcal{A}_1$.

Lemma 21 (Warm Start Guarantees)**.** *Let $\mathcal{A}_1$ be the output of* WARMSTART*, and let $\mathcal{E}_2$ be the event that (i) $\Delta_a \le 16\sigma\sqrt{2d/B}$ for all $a \in \mathcal{A}_1$, and (ii) $a^* \in \mathcal{A}_1$. Then $\mathbb{P}(\mathcal{E}_2 \mid \mathcal{E}_1) \ge 1 - \delta/3$.*

E.2.3 Step 3: Phased Elimination and Total Regret

With a warm-started active set $\mathcal{A}_1$ in hand, the main phased elimination loop of Algorithm 2 proceeds exactly as in the small-B regime, starting from $\mathcal{A}_1$ rather than $\mathcal{A}$.

We bound the Step 3 regret conditioned on $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$. Since $a^* \in \mathcal{A}_1$ by condition (ii) of $\mathcal{E}_2$, Lemma 17 continues to hold from epoch 1 onward.

Let R_{loc} denote the contribution to the regret from localization, and let $R_{\text{ref},h}$ denote the contribution from refinement in epoch h. For all epochs $h \ge 1$, since $\mathcal{A}_h \subseteq \mathcal{A}_1$, condition (i) of $\mathcal{E}_2$ guarantees that $\Delta_b \le 16\sigma\sqrt{2d/B}$ for all active arms. Therefore, the total localization regret across all H epochs is bounded by

$$R_{\text{loc}} \le 16\sigma\sqrt{\frac{2d}{B}}\sum_{h=1}^{H} m_{\text{loc},h} = \widetilde{O}\left(\sqrt{\frac{d}{B}} \cdot dB\log\left(\frac{1}{\delta}\right)\right) = \widetilde{O}\left(d^{3/2}\sqrt{B}\log\left(\frac{1}{\delta}\right)\right), \tag{42}$$

where we used the localization cost bound $\sum_h m_{\text{loc},h} = \widetilde{O}(dB\log(1/\delta))$ from (36) and $\sigma = \Theta(1)$. For the refinement regret in the first epoch ($h = 1$), we similarly use the warm-start gap bound:

$$R_{\text{ref},1} \le 16\sigma\sqrt{\frac{2d}{B}} \cdot m_{\text{ref},1} = \widetilde{O}\Big(d^{3/2}\sqrt{B}\,K\Big), \tag{43}$$

where we used (34). For epochs $h \ge 2$, since $a^* \in \mathcal{A}_1$, Lemma 18 applies and the argument is identical to that of Theorem 19 from epoch 2 onward, giving $\sum_{h=2}^{H} R_{\text{ref},h} = \widetilde{O}(\sqrt{KdT})$. Adding the localization and refinement components and substituting $K = \widetilde{O}(\log(|\mathcal{A}|/\delta))$, the total regret R_{phased} from the phased elimination part satisfies

$$R_{\text{phased}} = R_{\text{loc}} + R_{\text{ref},1} + \sum_{h=2}^{H} R_{\text{ref},h} = \widetilde{O}\left(d^{3/2}\sqrt{B}\log\Big(\frac{|\mathcal{A}|}{\delta}\Big) + \sqrt{dT\log\Big(\frac{|\mathcal{A}|}{\delta}\Big)}\right). \tag{44}$$

Total Regret. Combining R_{safe}, R_{ws}, and R_{phased}, conditioned on $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$,

$$R_T = \widetilde{O}\left(B\log\Big(\frac{|\mathcal{A}|}{\delta}\Big) + d^{3/2}\sqrt{B}\log^{5/2}\Big(\frac{|\mathcal{A}|}{\delta}\Big) + \sqrt{dT\log\Big(\frac{|\mathcal{A}|}{\delta}\Big)}\right).$$

It remains to show that $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$ holds with probability at least $1 - \delta$. Lemmas 20 and 21 give $\mathbb{P}(\mathcal{E}_1^c) \le \delta/3$ and $\mathbb{P}(\mathcal{E}_2^c \cap \mathcal{E}_1) \le \delta/3$. Lemma 15 gives $\mathbb{P}(\mathcal{E}_3^c \mid \mathcal{F}_3) \le \delta/3$, where $\mathcal{F}_3$ is the σ-algebra generated by the history before Step 3. Since $\mathcal{E}_1 \cap \mathcal{E}_2 \in \mathcal{F}_3$, it follows that $\mathbb{P}(\mathcal{E}_3^c \cap \mathcal{E}_1 \cap \mathcal{E}_2) \le \delta/3$. Therefore,

$$\mathbb{P}\big((\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3)^c\big) = \mathbb{P}(\mathcal{E}_1^c) + \mathbb{P}(\mathcal{E}_2^c \cap \mathcal{E}_1) + \mathbb{P}(\mathcal{E}_3^c \cap \mathcal{E}_1 \cap \mathcal{E}_2) \le \frac{\delta}{3} + \frac{\delta}{3} + \frac{\delta}{3} = \delta.$$

Hence, $\mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3) \ge 1 - \delta$ as desired.

E.3 Proofs of Auxiliary Lemmas

We now prove Lemmas 20 and 21, which establish the guarantees of Algorithms 3 and 4 respectively.

E.3.1 Proof of Lemma 20 (Safe Arm Guarantees)

Feasibility of arm pulls per batch. Set $m_\star = B/(2K_\star)$. Since

$$u_h(a) = \lceil B\rho_h(a)/(2K_\star)\rceil \le m_\star \rho_h(a) + 1,$$

we have

$$K_\star \sum_{a\in\mathrm{supp}(\rho_h)} u_h(a) \le K_\star(m_\star + |\,\mathrm{supp}(\rho_h)|) \le \frac{B}{2} + 4K_\star d(\log\log d + 11) \le B,$$

using $|\,\mathrm{supp}(\rho_h)| \le S_{\max}$ and the assumption on B. Any remaining slots are filled by an arbitrary arm whose rewards are ignored by the agent.

Concentration of arm estimates. Fix a round h and let $\mathcal{F}_{h-1}$ be the σ-algebra generated by the history immediately before its arm pulls, including $\mathcal{A}_h$, ρ_h, and $\{u_h(a)\}$. Condition on a fixed realization of this history on which the round's batch can be completed by time T. Since each block j uses fresh arm pulls with the same fixed protocol, the $K_\star$ block estimates $b^\top\hat\theta_h^{(1)}, \dots, b^\top\hat\theta_h^{(K_\star)}$ are i.i.d. for any fixed $b \in \mathcal{A}_h$.

Because $\bar Y_{a,h}^{(j)}$ is the sample mean of $u_h(a)$ independent rewards, $\mathbb{E}[\bar Y_{a,h}^{(j)}] = \mu_a$ and $\mathrm{Var}(\bar Y_{a,h}^{(j)}) \le \sigma^2/u_h(a)$. Since $\mathbb{E}[b^\top\hat\theta_h^{(j)}] = \mu_b$ (no bias), the variance of the block estimate satisfies

$$\mathrm{Var}(b^\top\hat\theta_h^{(j)}) \le \sum_{a\in\mathrm{supp}(\rho_h)} (b^\top G_h^{-1}a)^2\rho_h(a)^2 \cdot \frac{\sigma^2}{u_h(a)} \le \frac{\sigma^2}{m_\star}\sum_{a\in\mathrm{supp}(\rho_h)} (b^\top G_h^{-1}a)^2\rho_h(a) \le \frac{2d\sigma^2}{m_\star},$$

using $u_h(a) = \lceil m_\star\rho_h(a)\rceil \ge m_\star\rho_h(a)$ and Lemma 6.

Applying Lemma 5 to the $K_\star$ i.i.d. block estimates with variance at most $2d\sigma^2/m_\star$, using failure probability $\delta/(3|\mathcal{A}|\lceil\log_2|\mathcal{A}|\rceil)$ and substituting $K_\star m_\star = B/2$, we have

$$|\hat\mu_{b,h} - \mu_b| \le \sqrt{\frac{64d\sigma^2}{K_\star m_\star}\log\Big(\frac{3|\mathcal{A}|\lceil\log_2|\mathcal{A}|\rceil}{\delta}\Big)} = 8\sigma\sqrt{\frac{2d}{B}\log\Big(\frac{3|\mathcal{A}|\lceil\log_2|\mathcal{A}|\rceil}{\delta}\Big)} =: \varepsilon_{\mathrm{safe}},$$

with probability at least $1 - \delta/(3|\mathcal{A}|\lceil\log_2|\mathcal{A}|\rceil)$ under our conditioning. A union bound over $b \in \mathcal{A}_h$, using $|\mathcal{A}_h| \le |\mathcal{A}|$, gives

$$\mathbb{P}\Big(E_h \;\Big|\; \mathcal{F}_{h-1}\Big) \le \frac{\delta}{3\lceil\log_2|\mathcal{A}|\rceil} \quad \text{where} \quad E_h := \{\exists\, b \in \mathcal{A}_h : |\hat\mu_{b,h} - \mu_b| > \varepsilon_{\mathrm{safe}}\}.$$

Define

$$\mathcal{E}_{\mathrm{safe}} := \bigcap_{h=1}^{\lceil\log_2|\mathcal{A}|\rceil} (E_h)^c = \bigcap_{h=1}^{\lceil\log_2|\mathcal{A}|\rceil}\ \bigcap_{b\in\mathcal{A}_h} \{|\hat\mu_{b,h} - \mu_b| \le \varepsilon_{\mathrm{safe}}\}.$$

Taking a union bound over rounds and applying the tower property give

$$\mathbb{P}(\mathcal{E}_{\mathrm{safe}}^c) = \mathbb{P}\left(\bigcup_{h=1}^{\lceil\log_2|\mathcal{A}|\rceil} E_h\right) \le \sum_{h=1}^{\lceil\log_2|\mathcal{A}|\rceil} \mathbb{P}(E_h) = \sum_{h=1}^{\lceil\log_2|\mathcal{A}|\rceil} \mathbb{E}\left[\mathbb{P}\Big(E_h \;\Big|\; \mathcal{F}_{h-1}\Big)\right] \le \frac{\delta}{3}.$$

Suboptimality gap of the safe arm. On completion of SAFEARM, condition on $\mathcal{E}_{\mathrm{safe}}$. In round h, let $a_h^* = \arg\max_{a\in\mathcal{A}_h}\mu_a$. If the group containing a_h^* is eliminated, there exists b in the retained group with $\hat\mu_{b,h} \ge \hat\mu_{a_h^*,h}$, so

$$\mu_b \ge \hat\mu_{b,h} - \varepsilon_{\mathrm{safe}} \ge \hat\mu_{a_h^*,h} - \varepsilon_{\mathrm{safe}} \ge \mu_{a_h^*} - 2\varepsilon_{\mathrm{safe}}.$$

Hence $\mu_{a_{h+1}^*} \ge \mu_{a_h^*} - 2\varepsilon_{\mathrm{safe}}$, and induction from $\mu_{a_1^*} = \mu_{a^*}$ yields

$$\mu_{a_{h+1}^*} \ge \mu_{a^*} - 2h\,\varepsilon_{\mathrm{safe}}.$$

In other words, $\Delta_{a_{h+1}^*} \le 2h\,\varepsilon_{\mathrm{safe}}$. When the algorithm returns, at most $\lceil\log_2|\mathcal{A}|\rceil$ rounds have been completed, and the active set contains only a single arm a_{safe}. Applying the induction bound gives $\Delta_{a_{\mathrm{safe}}} \le 2\lceil\log_2|\mathcal{A}|\rceil\,\varepsilon_{\mathrm{safe}}$. Substituting $\varepsilon_{\mathrm{safe}}$ yields (40), i.e., event $\mathcal{E}_1$ holds. It follows that $\mathbb{P}(\mathcal{E}_1) \ge \mathbb{P}(\mathcal{E}_{\mathrm{safe}}) \ge 1 - \delta/3$.

E.3.2 Proof of Lemma 21 (Warm Start Guarantees)

Let $\mathcal{F}_{\mathrm{ws}}$ be the σ-algebra generated by the history immediately before Algorithm 4. In Parts 1–3 below, we condition on a fixed entry history, including the realized arm a_{safe}; probabilities in those parts refer to this conditional law. Localization and refinement observations are fresh, and filler rewards are not used in the estimates. We establish $\mathbb{P}(\mathcal{E}_{\mathrm{ws}}^c \mid \mathcal{F}_{\mathrm{ws}}) \le \delta/3$ uniformly over these histories, where $\mathcal{E}_{\mathrm{ws}} := \mathcal{E}_{\mathrm{ws,loc}} \cap \mathcal{E}_{\mathrm{ws,ref}}$ (see below for the definitions).

Part 1: Localization event $\mathcal{E}_{\mathrm{ws,loc}}$. For each $a \in \mathrm{supp}(\rho_0)$, 1BitLocalize is called with effective standard deviation $\sigma' = \sigma/\sqrt{u_0(a)}$ and failure probability $\delta/(6S_{\max})$. Since $|\mathrm{supp}(\rho_0)| \le S_{\max}$, Proposition 7 and a union bound give that

$$\mathcal{E}_{\mathrm{ws,loc}} := \bigcap_{a \in \mathrm{supp}(\rho_0)} \left\{ \mu_a \in [L_a, U_a] \text{ and } U_a - L_a \le \frac{8\sigma}{\sqrt{u_0(a)}} \right\} \tag{45}$$

has conditional probability $\mathbb{P}(\mathcal{E}_{\mathrm{ws,loc}} \mid \mathcal{F}_{\mathrm{ws}}) \ge 1 - \delta/6$.

Part 2: Refinement event $\mathcal{E}_{\mathrm{ws,ref}}$. Let $\mathcal{G}_0$ be the σ-algebra generated by the history after all localization calls in Algorithm 4. Condition on a fixed realization of $\mathcal{G}_0$ in which localization is completed, all its intervals are valid (i.e., $\mathcal{E}_{\mathrm{ws,loc}}$ holds), and enough pulls remain to complete the allocated refinement stage. All expectations, biases, and variances below refer to the conditional law given $\mathcal{G}_0$. Since each block j uses independent arm pulls with the same fixed parameters, the K_0 estimates $b^\top \hat{\theta}_0^{(1)}, \dots, b^\top \hat{\theta}_0^{(K_0)}$ are i.i.d. for any fixed $b \in \mathcal{A}$. We bound their variance and bias as follows:

- *Variance.* Each call to 1BitRefine for arm a uses $n_{\mathrm{q}} = \lceil C_V \log_2 T \rceil$ queries. Because arm a is pulled only $u_0(a)$ times per query (with the rest of the batch padded by a_{safe}), the effective standard deviation is $\sigma' = \sigma/\sqrt{u_0(a)}$. Applying the bound from Corollary 5 yields

$$\mathrm{Var}(\hat{Y}_{a,0}^{(j)}) \le \frac{C_V(\sigma')^2 \log_2 T}{n_{\mathrm{q}}} \le \frac{C_V(\sigma^2/u_0(a)) \log_2 T}{C_V \log_2 T} = \frac{\sigma^2}{u_0(a)}.$$

  Combining this with $u_0(a) \ge B\rho_0(a)$ and Lemma 6, we obtain

$$\mathrm{Var}(b^\top \hat{\theta}_0^{(j)}) = \sum_a (b^\top G_0^{-1} a)^2 \rho_0(a)^2 \cdot \mathrm{Var}(\hat{Y}_{a,0}^{(j)}) \le \frac{\sigma^2}{B} \sum_a (b^\top G_0^{-1} a)^2 \rho_0(a) \le \frac{2d\sigma^2}{B}.$$

- *Bias.* Corollary 5 gives $|\mathrm{Bias}(\hat{Y}_{a,0}^{(j)})| \le 3\sigma/(T^2\sqrt{u_0(a)}) \le 3\sigma/T^2$. By writing $\mu_b = b^\top G_0^{-1} G_0 \theta$ and using the triangle inequality alongside Corollary 4, we obtain

$$|\mathbb{E}[b^\top \hat{\theta}_0^{(j)}] - \mu_b| \le \frac{3\sigma}{T^2} \sum_{a \in \mathrm{supp}(\rho_0)} \rho_0(a) |b^\top G_0^{-1} a| \le \frac{3\sqrt{2d}\,\sigma}{T^2} \le 2\sigma\sqrt{\frac{2d}{B}},$$

  where the last step uses $T \ge B$ and $B \ge 4d(\log\log d + 11) \ge 4$, giving $T^2 \ge B^2 \ge 4B \ge \frac{3}{2}\sqrt{B}$.

Applying Lemma 5 to the K_0 i.i.d. block estimates with variance at most $2d\sigma^2/B$ and failure probability $\delta/(6|\mathcal{A}|)$, then combining with the bias bound via the triangle inequality, we obtain

$$|\hat{\mu}_{b,0} - \mu_b| \le \sqrt{\frac{64d\sigma^2}{BK_0} \log\Big(\frac{6|\mathcal{A}|}{\delta}\Big)} + 2\sigma\sqrt{\frac{2d}{B}} \le 4\sigma\sqrt{\frac{2d}{B}},$$

with probability at least $1 - \delta/(6|\mathcal{A}|)$ under our conditioning, where the last step uses $K_0 = \lceil 8\log(6|\mathcal{A}|/\delta) \rceil$. Since $\mathcal{A}$ is fixed, a union bound over all $b \in \mathcal{A}$ gives

$$\mathbb{P}\Big(\mathcal{E}_{\mathrm{ws,ref}}^c \,\Big|\, \mathcal{G}_0\Big) \le \frac{\delta}{6} \quad \text{where} \quad \mathcal{E}_{\mathrm{ws,ref}} := \bigcap_{b \in \mathcal{A}} \left\{ |\hat{\mu}_{b,0} - \mu_b| \le 4\sigma\sqrt{\frac{2d}{B}} \right\}.$$

Since $\mathcal{F}_{\mathrm{ws}} \subseteq \mathcal{G}_0$ and $\mathcal{E}_{\mathrm{ws,loc}} \in \mathcal{G}_0$, the tower property gives

$$\mathbb{P}(\mathcal{E}_{\mathrm{ws,ref}}^c \cap \mathcal{E}_{\mathrm{ws,loc}} \mid \mathcal{F}_{\mathrm{ws}}) = \mathbb{E}\left[\mathbb{1}\{\mathcal{E}_{\mathrm{ws,loc}}\} \cdot \mathbb{P}(\mathcal{E}_{\mathrm{ws,ref}}^c \mid \mathcal{G}_0) \,\middle|\, \mathcal{F}_{\mathrm{ws}}\right] \le \frac{\delta}{6}.$$

Part 3: Optimality of retained arms.

On completion of WARMSTART, under $\mathcal{E}_{\text{ws}} = \mathcal{E}_{\text{ws,loc}} \cap \mathcal{E}_{\text{ws,ref}}$, we have $|\hat{\mu}_{b,0} - \mu_b| \le 4\sigma\sqrt{2d/B}$ for all $b \in \mathcal{A}$. We then have the following two conditions.

Condition (i): Gap bound. For any $a \in \mathcal{A}_1$, the elimination rule (see Line 15) gives

$$\hat{\mu}_{a^*,0} - \hat{\mu}_{a,0} \le 8\sigma\sqrt{2d/B}.$$

Combining this with standard decomposition and the estimation error guarantee under $\mathcal{E}_{\text{ws,ref}}$ yields

$$\Delta_a = \mu_{a^*} - \mu_a \le \underbrace{(\mu_{a^*} - \hat{\mu}_{a^*,0})}_{\le 4\sigma\sqrt{2d/B} \text{ under } \mathcal{E}_{\text{ws,ref}}} + (\hat{\mu}_{a^*,0} - \hat{\mu}_{a,0}) + \underbrace{(\hat{\mu}_{a,0} - \mu_a)}_{\le 4\sigma\sqrt{2d/B} \text{ under } \mathcal{E}_{\text{ws,ref}}} \le 16\sigma\sqrt{\frac{2d}{B}}.$$

Condition (ii): Optimal arm retained. If $a^* \notin \mathcal{A}_1$, some b satisfies $\hat{\mu}_{b,0} - \hat{\mu}_{a^*,0} > 8\sigma\sqrt{2d/B}$, which gives $\mu_b - \mu_{a^*} > 8\sigma\sqrt{2d/B} - 2 \cdot 4\sigma\sqrt{2d/B} = 0$, contradicting the optimality of a^*. Thus, it must hold that $a^* \in \mathcal{A}_1$.

Part 4: Conclusion. Parts 1–2 give

$$\mathbb{P}(\mathcal{E}_{\text{ws}}^c \mid \mathcal{F}_{\text{ws}}) \le \mathbb{P}(\mathcal{E}_{\text{ws,loc}}^c \mid \mathcal{F}_{\text{ws}}) + \mathbb{P}(\mathcal{E}_{\text{ws,ref}}^c \cap \mathcal{E}_{\text{ws,loc}} \mid \mathcal{F}_{\text{ws}}) \le \frac{\delta}{6} + \frac{\delta}{6} = \frac{\delta}{3}.$$

Since $\mathcal{E}_{\text{ws}} \subseteq \mathcal{E}_2$ and $\mathcal{E}_1 \in \mathcal{F}_{\text{ws}}$, this uniform conditional bound implies $\mathbb{P}(\mathcal{E}_2 \mid \mathcal{E}_1) \ge 1 - \delta/3$, as desired.

E.4 Proof of Corollary 3 (Gap to The Lower Bound)

In this section, we provide the analysis for Corollary 3. For $d \le |\mathcal{A}| = \text{poly}(d)$ and $\sigma = \Theta(1)$, we show that the expected regret of Algorithm 2 is order-optimal up to $\text{polylog}(d, T)$ factors provided that either $B = \Omega(d^3)$ or $T = \Omega(d^2 B)$. Furthermore, in all remaining regimes, we show that its expected regret is within a $\widetilde{O}(\sqrt{d})$ factor of the minimax lower bound implied by Corollary 1. Throughout, $\widetilde{O}(\cdot)$ hides $\text{polylog}(d, T)$ factors, even when d and T do not appear explicitly in its argument. Write

$$\text{LB}(B, T, d) := \min\{T,\ B + \sqrt{dT}\}.$$

Corollary 1 gives a minimax lower bound of $\widetilde{\Omega}(\text{LB}(B, T, d))$. If $\text{LB}(B, T, d) = T$, then the trivial upper bound $\mathbb{E}[R_T] \le 2T$ is already optimal to within logarithmic factors. Hence, below we assume $\text{LB}(B, T, d) = B + \sqrt{dT} < T$, and we choose $\delta = 1/T$.

We first recall the simplified expected regret bound of Corollary 2, under the assumptions that $|\mathcal{A}| = \text{poly}(d)$ and $\sigma = \Theta(1)$:

$$\mathbb{E}[R_T] = \widetilde{O}\left(B + d^{3/2}\sqrt{B} + \sqrt{dT}\right). \tag{46}$$

The middle term is $O(B)$ when $B = \Omega(d^3)$, and $O(\sqrt{dT})$ when $T = \Omega(d^2 B)$. Therefore, either condition gives $\mathbb{E}[R_T] = \widetilde{O}(B + \sqrt{dT})$. We now bound the gap in the remaining regimes according to whether the algorithm includes the SAFEARM and WARMSTART stages.

Case 1: Without warm start. In this case, $B = \widetilde{O}(d)$ by Line 3 of Algorithm 2. Theorem 19 gives

$$\mathbb{E}[R_T] = \widetilde{O}\left(dB + \sqrt{dT}\right),$$

since the failure event contributes only $O(T\delta) = O(1)$ to expected regret. If $T \le dB$, the trivial bound $\mathbb{E}[R_T] \le 2T$ gives

$$\frac{\mathbb{E}[R_T]}{\text{LB}(B, T, d)} = \widetilde{O}\left(\frac{T}{B + \sqrt{dT}}\right) = \widetilde{O}\left(\frac{T}{\sqrt{dT}}\right) = \widetilde{O}\left(\sqrt{\frac{T}{d}}\right) = \widetilde{O}(\sqrt{B}) = \widetilde{O}(\sqrt{d}).$$

If $T > dB$, then $dB \le \sqrt{B}\sqrt{dT}$. It follows that

$$\frac{\mathbb{E}[R_T]}{\text{LB}(B, T, d)} = \widetilde{O}\left(\frac{dB + \sqrt{dT}}{B + \sqrt{dT}}\right) = \widetilde{O}\left(\frac{\sqrt{B}\sqrt{dT}}{\sqrt{dT}}\right) = \widetilde{O}(\sqrt{B}) = \widetilde{O}(\sqrt{d}).$$

Case 2: With warm start. The pulls in SAFEARM and the core-set pulls in WARMSTART together contribute at most $\widetilde{O}(B)$ regret. All other pulls use either a_{safe} during WARMSTART or an arm in $\mathcal{A}_1$ during Step 3, since subsequent active sets are subsets of $\mathcal{A}_1$. On $\mathcal{E}_1 \cap \mathcal{E}_2$, these arms have gap $\widetilde{O}(\sqrt{d/B})$ by Lemmas 20 and 21. The two lemmas also give $\mathbb{P}((\mathcal{E}_1 \cap \mathcal{E}_2)^c) \le 2/(3T)$, so the failure event contributes only $O(1)$ to expected regret. Hence, including any interrupted stage, we have

$$\mathbb{E}[R_T] = \widetilde{O}\left(B + T\sqrt{\frac{d}{B}}\right).$$

If $T \le dB$, the preceding bound gives the following (via $\sqrt{T} \le \sqrt{dB}$):

$$\mathbb{E}[R_T] = \widetilde{O}(B + d\sqrt{T}),$$

If $T > dB$, the general upper bound in (46) gives the same conclusion, since $d^{3/2}\sqrt{B} \le d\sqrt{T}$ and $\sqrt{dT} \le d\sqrt{T}$. Therefore,

$$\frac{\mathbb{E}[R_T]}{\mathrm{LB}(B,T,d)} = \widetilde{O}\left(\frac{B + d\sqrt{T}}{B + \sqrt{dT}}\right) = \widetilde{O}(\sqrt{d}).$$

The ratio is therefore $\widetilde{O}(\sqrt{d})$ in both cases. This completes the proof of Corollary 3.

F Discussion on Downlink Communication

While our problem formulation does not limit the amount of downlink communication, it is worth noting what needs to be sent from the learner to the agent in each of our algorithms. For Algorithm 1, the amount of downlink communication is relatively small; in each batch, the learner sends (i) a single arm to be played, and (ii) a single threshold query specified by its threshold and direction (i.e., $\ge$ or $\le$).[14]

Algorithm 2 has similar communication requirements in the phased elimination part, but it incurs much higher communication costs in the pre-processing steps described in Section 4.3. Perhaps most significantly, for safe-arm identification (Algorithm 3), the learner splits the set of non-eliminated actions in half and reveals the two halves to the agent; a direct implementation thus incurs $\Omega(|\mathcal{A}|)$ overhead in the early rounds. Moreover, the agent needs to compute $\hat{\theta}_h^{(j)}$ in (8), which requires knowing the allocation ρ_h. In contrast, the requirements for the warm-start procedure (Algorithm 4) are milder, with each batch only specifying two arm identifiers, their prescribed allocations, and a threshold query for localization or refinement.

We also note that our analysis assumes that the agent is given exact real numbers (e.g., a threshold). One could also consider encoding these to $1/\mathrm{poly}(T)$ accuracy using $O(\log T)$ bits, but studying the effect of such approximations is beyond our scope. Overall, we leave it for future work to study settings in which the downlink communication is also constrained.

G Extension to Multiple Bits per Batch (Remark 5)

We now extend our main results to the q-bit setting, where $q \ge 1$ is an integer and the agent may transmit up to q bits of feedback per batch. Each batch still contains B pulls, and all feedback is returned only at its end.

Lower bound. Before batch i, the feedback transcript contains at most $q(i-1)$ bits. Using this bound in the proof of Theorem 1, together with the first-batch penalty, gives the communication-based lower bound

$$\Omega\left(\min\left\{T,\ B + \frac{B}{q}\min\{d, \log|\mathcal{A}|\}\right\}\right).$$

The $\Omega(B)$ penalty remains even with unquantized feedback, since the actions in the first batch are chosen before any rewards are observed.

[14] In fact, $\ge$ comparisons alone suffice. The localization procedure of (Lau and Scarlett, 2026c, Appendix A, Step 1) can be implemented in this form, and the $<$ responses in (13) are obtained by complementing the corresponding $\ge$ bits.

Upper-bound constructions. The high-level idea is to divide each batch of B pulls into q sub-batches of $b = B/q$ pulls each,[15] allowing one query per sub-batch (and hence q queries per batch). However, these sub-batches must be handled in parallel: all their actions and queries must be fixed before the batch begins, since feedback arrives only at the end of the batch. We therefore adapt the 1-bit algorithms to use effective batch size b. A high-level outline is given as follows, and further details are given below:

- *Refinement.* Once the localization intervals are fixed, the refinement queries in Appendix B.2 are non-adaptive. We can therefore include queries from different arms and median-of-means blocks in the same batch, without further modification to the estimator.
- *Localization.* We use the localizer based on noisy binary search in Appendix B.1. Queries for different arms can share a batch, but successive queries for the same arm require feedback from earlier batches. If fewer than q arms still require localization queries, some sub-batches are unused for localization.[16] We account for the resulting additional regret below.
- *Warm start.* We use the same localization and refinement modifications as above. The remaining pulls within each *sub-batch* are allocated to a_{safe}, as in Lines 6 and 9 of Algorithm 4. The guarantees for the retained arms in Lemma 21 hold with b in place of B, up to absolute constants.
- *Safe arm identification.* We replace each binary comparison in Algorithm 3 by a comparison among up to 2^q groups of candidate arms, communicating the winning group's index.

Throughout, the estimation parameters and allocations use b in place of B. Line 3 of Algorithm 2 also uses b to select its small- or large-batch branch. The only exception is SAFEARM, whose comparisons use a full batch of B pulls. We now proceed to describe the above ideas in more detail.

Refinement and localization. In each phased elimination epoch, each sub-batch contains b pulls of a single arm. As in the full-batch construction in Appendix B, the query uses their average, giving effective standard deviation $\sigma' = \sigma/\sqrt{b}$. We retain the original failure probabilities and use disjoint fresh observations for all queries.

For refinement, we combine queries across arms and median-of-means blocks into batches of q queries, with at most one batch containing fewer queries per refinement stage. We allocate any unused sub-batches to arbitrary active arms and discard their rewards, adding at most $2B$ regret per refinement stage.

For localization, each batch makes one query for each of q different arms whose localization is unfinished, using one sub-batch per arm. To preserve the localization protocol in Appendix B.1, each arm's next query is chosen using its responses from earlier batches. If fewer than q arms still require localization queries, we use one sub-batch per such arm for its next query in the current batch. Further queries for these arms must wait for feedback at the end of the batch, so we allocate the other sub-batches to arbitrary active arms and discard their rewards. Since every unfinished arm receives one query in each such batch, the number of batches with sub-batches unused for localization is at most the largest number of localization queries needed by any arm. Since each such batch incurs at most $2B$ regret, Proposition 7 gives $O\Big(B\left[\log(\sqrt{b}/\sigma) + \log(1/\delta_{\text{loc}})\right]\Big)$ additional regret per localization stage.

Summing the additional regret from refinement and localization over the $O(\log T)$ epochs and the warm-start stage gives $\widetilde{O}(B \log T \log(1/\delta))$ additional regret in total.

Safe arm identification. We use the same core-set sampling and estimation procedures as in Algorithm 3, with B pulls per comparison to estimate the means of all current candidate arms. However, successive binary comparisons require separate batches, since the surviving candidate set is determined by feedback received only at the end of the current batch. Accordingly, in order to

[15] For simplicity, assume $q \le B$ and q divides B. For general q, we allocate $\bar{q} = \min\{q, B\}$ sub-batches within a batch, each containing $b = \lfloor B/\bar{q} \rfloor$ pulls. The remaining $B - \bar{q}b < \bar{q}b$ pulls arise from rounding and are allocated by repeating a subset of the planned pulls exactly once. Their rewards are discarded, so no additional queries are made. This at most doubles the regret. Since $b = \Theta(B/\bar{q})$, the upper bounds hold with q replaced by $\bar{q}$, up to constant factors.

[16] Using the non-adaptive localizer of (Lau and Scarlett, 2026a, Lemma 1) would allow multiple queries for the same arm to share a batch, simplifying the scheduling and the additional regret analysis. It matches the interval-width and query-complexity bounds of Proposition 7 up to absolute constants, but uses general binary queries rather than simple threshold comparisons.

make full use of the q bits available, we instead partition the current candidates into $\min\{2^q, |\mathcal{A}_h|\}$ balanced groups at each comparison. The agent now sends the index of the group containing an empirical maximizer using at most q bits (rather than one bit selecting between two groups). After at most $\lceil (\log_2 |\mathcal{A}|)/q \rceil$ comparisons, one candidate remains, giving regret

$$O\left(B\left\lceil \frac{\log_2 |\mathcal{A}|}{q} \right\rceil\right) = O\left(B + b \log |\mathcal{A}|\right).$$

The simultaneous confidence bounds in Appendix E.3.1 still apply, so retaining the group containing an empirical maximizer gives the same per-comparison guarantee. There are no more comparisons than before, so the original guarantee of Lemma 20 continues to hold. Since $b \le B$, this also implies its returned-arm gap bound with b in place of B and the same failure probability, which suffices for the subsequent warm-start regret analysis.

Resulting regret bounds. The regret analyses for localization, refinement, and WARMSTART in Appendices D and E apply to the queried sub-batches with B replaced by b. Adding the $\widetilde{O}(B \cdot \log T \cdot \log(1/\delta))$ overhead described above and the modified SAFEARM regret for Algorithm 2 gives the following bounds:

- Under the assumptions of Theorem 3, the modified Algorithm 1 satisfies, with probability at least $1 - \delta$,
$$R_T = \widetilde{O}\left((B + db) \log \frac{1}{\delta} + d\sqrt{T \log \frac{1}{\delta}}\right).$$
- Under the assumptions of Theorem 4, the modified Algorithm 2 satisfies, with probability at least $1 - \delta$,
$$R_T = \widetilde{O}\left(B \log \frac{1}{\delta} + b \log \frac{|\mathcal{A}|}{\delta} + d^{3/2}\sqrt{b} \log^{5/2} \frac{|\mathcal{A}|}{\delta} + \sqrt{dT \log \frac{|\mathcal{A}|}{\delta}}\right).$$

Substituting $b = B/q$ then gives the bounds stated in Remark 5.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract gives a brief outline of the problem setup, alongside an outline of our main results. Similarly, the introduction gives a more detailed exposition of the problem, alongside comparisons with similar works in the literature. We also forward-reference the detailed problem setup and main results in the introduction, with these sections building on the discussions in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: The limitations of the paper are discussed in Section 3.3. Additionally, we discuss possible future directions of work that address these limitations in the conclusion.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: A detailed problem setup (including all relevant assumptions) is given in Section 2. Detailed proofs of Theorems 1 and 2 are given in Appendix C, while the proofs of Theorems 3 and 4 are given in Appendices D and E respectively.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [N/A]

   Justification: The paper does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [N/A]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [N/A]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [N/A]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Due to the theoretical nature of this work, there are no ethical issues concerning human participation. Similarly, there are no data-related concerns, since no datasets are used. Regarding societal impact, this paper does not violate any of the discussed areas of concern, and our work has no direct relation to any specific individuals or any gender, race, sexuality, or religion.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: We do not anticipate any direct societal impact of the work performed, particularly given its theoretical nature.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We believe that our work poses no safeguarding risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [N/A]

Justification: This paper uses no existing assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: This paper does not release any new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were used as supporting tools for manuscript preparation and technical revision, rather than as components of the research methodology. See the Declaration of LLM Usage for details.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2026/LLM) for what should or should not be described.