Proceedings of Machine Learning Research vol 247:1–48, 2024 37th Annual Conference on Learning Theory

The Best Arm Evades: Near-optimal Multi-pass Streaming Lower Bounds for Pure Exploration in Multi-armed Bandits

Sepehr Assadi SEPEHR@ASSADI.INFO
University of Waterloo

Chen Wang CW200@RICE.EDU
Rice University and Texas A&M University

Editors: Shipra Agrawal and Aaron Roth

Abstract

We give a near-optimal sample-pass trade-off for pure exploration in multi-armed bandits (MABs) via multi-pass streaming algorithms: any streaming algorithm with sublinear memory that uses the optimal sample complexity of $O(n/\Delta^2)$ requires $\Omega(\log(1/\Delta)/\log\log(1/\Delta))$ passes. Here, n is the number of arms and Δ is the reward gap between the best and the second-best arms. Our result matches the $O(\log(1/\Delta))$ pass algorithm of Jin et al. [ICML'21] (up to lower order terms) that only uses $O(1)$ memory and answers an open question posed by Assadi and Wang [STOC'20].

1. Introduction

Pure exploration in multi-armed bandits (MABs) is a fundamental problem in machine learning (ML) and theoretical computer science (TCS). The classical setting of the problem is as follows: we are given n arms with unknown sub-Gaussian reward distributions, and we want to find the best arm, defined as the arm with the highest expected reward, with a high probability and a small number of arm pulls. The problem has been extensively studied in the learning theory community (e.g. Even-Dar et al. (2002); Mannor and Tsitsiklis (2004); Kalyanakrishnan and Stone (2010); Agarwal et al. (2017); Chen et al. (2017)), and it has found applications in various fields like online advertisement Bertsimas and Mersereau (2007); Schwartz et al. (2017), clinical trials Villar et al. (2015); Aziz et al. (2021), content optimization Agarwal et al. (2009); Li et al. (2010), among others.

Under the classical (RAM) setting, the sample complexity of $\Theta(\frac{n}{\Delta^2})$ is shown to be necessary and sufficient to find the best arm with high constant probability (Even-Dar et al. (2002); Even-Dar et al. (2006); Mannor and Tsitsiklis (2004), cf. Karnin et al. (2013); Jamieson et al. (2014)). Here, and throughout, Δ is used to denote the gap between the mean of the best and the second-best arms. On the flip side, all the classical algorithms require the entire set of arms to be stored for repeated visit. In modern large-scale applications, such a memory requirement may render the algorithms inefficient. In light of this, Assadi and Wang (2020) introduced the *streaming* multi-armed bandits model, in which the arms arrive one-by-one in a stream, and the algorithm is only allowed to store $o(n)$ arms, and ideally much smaller, like $O(1)$ or $\text{polylog}(n)$ arms, at any time. Perhaps surprisingly, Assadi and Wang (2020) showed that if we are given the value of Δ *a priori*, there exists a single-pass streaming algorithm that finds the best arm with high constant probability, uses $O(\frac{n}{\Delta^2})$ samples, and only maintains a memory of a single extra arm.

The results of Assadi and Wang (2020) led to a nascent line of work on MABs in the streaming modelMaiti et al. (2021); Jin et al. (2021); Assadi and Wang (2022); Agarwal et al. (2022); Wang (2023); Li et al. (2023). For the pure exploration problem, it has been shown by Assadi and Wang (2022) that the prior knowledge of Δ is necessary for the single-pass algorithm: if this piece of information is not available and only the stream of arms itself is provided, then any single-pass algorithm with $o(n)$-arm memory that finds the best arm with high constant probability can incur an *unbounded* sample complexity (as a function of Δ). On the positive side, Jin et al. (2021) designs an algorithm with $O(\frac{n}{\Delta^2})$ sample complexity and a memory of a single arm in $\log(\frac{1}{\Delta})$ passes, even if the knowledge of Δ is *not* given a priori [1]. This large gap between the positive results with $\log(\frac{1}{\Delta})$ passes and the lower bound in the single-pass setting presents us with the exciting open question:

If no additional knowledge is given apart from the stream, how many passes are ***necessary*** *for streaming MABs algorithms with $o(n)$-arm memory to find the best arm with $O(\frac{n}{\Delta^2})$ samples?*

The open question was initially mentioned by Assadi and Wang (2020) and was later re-formulated in Assadi and Wang (2022)[2]. The quest of the tight pass bound is similar-in-spirit to the lower bounds in collaborative learning Tao et al. (2019) and multi-pass *regret minimization* MABs Agarwal et al. (2022); however, none of the techniques in the aforementioned lower bounds can be directly used for this problem (see Section 1.1 for a detailed discussion), which renders the open problem even more fascinating.

In this work, we provide the answer to the open question: we show that (almost) $\Omega(\log(1/\Delta))$ passes are *necessary* (up to exponentially smaller factors) for any algorithms with $o(n)$ memory to find the best arm. More formally, our main result can be presented as follows.

Result 1 (Informal version of Corollary 2) *Any streaming algorithm that finds the best arm with probability at least $\frac{1999}{2000}$ using a memory of $o\left(\frac{n}{\log^3(1/\Delta)}\right)$ arms and a sample complexity of $C \cdot \frac{n}{\Delta^2}$ for any constant C has to use $\Omega\left(\frac{\log(\frac{1}{\Delta})}{\log\log(\frac{1}{\Delta})}\right)$ passes.*

Our lower bound asymptotically matches the pass bound of the algorithm in Jin et al. (2021) up to the exponentially smaller $O(\log\log(1/\Delta))$ term. Furthermore, as long as $\Delta \geqslant 1/2^{n^{1/3-\Omega(1)}}$, which is a quite natural assumption, our result demonstrates a sharp dichotomy on the pass-memory trade-off: if we use slightly less than $O(\log(1/\Delta))$ passes, no algorithm with even slightly less than n-arm memory can succeed with a good probability; however, if we slightly increase the number of passes to $O(\log(1/\Delta))$, it is possible to find the best arm with high constant probability with only a single arm of memory.

1.1. Our Techniques

The proof of our result is based on a novel inductive argument that explicitly keeps track of the information revealed to the algorithm in each pass. This is in sharp contrast with all other

1. The algorithm actually achieves a stronger instance-sensitive sample complexity – see Section 1.2 for a discussion.
2. The problem is discussed at multiple open problem sessions of conferences and workshops, e.g, WALD(O) 2021.

lower bounds that address 'rounds' or 'passes' for MABs in similar contexts (e.g., Agarwal et al. (2017); Tao et al. (2019); Karpov et al. (2020); Agarwal et al. (2022); Karpov and Zhang (2023)) that are based on *round/pass elimination* ideas. To elaborate, let us take a closer look at Agarwal et al. (2022), which studied multi-pass streaming lower bounds in MABs for regret minimization. Roughly speaking, both Agarwal et al. (2022) and our lower bound instances divide the stream into equal-sized *batches*. Each batch contains a single arm with mean reward either $\frac{1}{2}$ or $> \frac{1}{2}$, and the rest of the arms in the batch have mean reward $\frac{1}{2}$. The *intuition* here is that by arranging the batches that *may* have higher mean rewards to arrive later, the algorithm is forced to be 'conservative' at each pass to only 'eliminate' the last batch. To this end, the main technical step of Agarwal et al. (2022) is to reduce proving the lower bound for P-pass algorithms to proving a lower bound for $(P-1)$-pass algorithms—this is the so-called round/pass elimination idea. However, as the algorithm can gain information in each pass, the instance distribution from the algorithm's internal view is inevitably 'more biased'. As such, a key part of the analysis in Agarwal et al. (2022) is a delicate handling of the change in the distribution of instances from one round to the next, and ensures the change is not too much.

For our purpose, round elimination seems to ask too much from the argument to make sure that the distribution only slightly changes. As such, we proceed differently by *allowing* the instance distribution to significantly change between rounds. Concretely, for a P-pass algorithm, we divide the arms into $(P+1)$ equal-sized batches, and arrange them in the *reversed* order of the stream arrival, i.e., the stream is composed of $(B_{P+1}, B_P, \cdots, B_1)$. Each batch *may* contain an arm with mean reward $\frac{1}{2}+\eta_p$, and the rest of the arms are 'flat', i.e., with mean reward $\frac{1}{2}$. The parameter η_p decreases by a polynomial factor of $1/P$, i.e., $\eta_{p+1} \leqslant (1/P^{15}) \cdot \eta_p$. At each pass p, we show that the algorithm so far has not gained enough "knowledge" about the batch B_p such that even if the algorithm knows that none of the batches $B_1, \ldots, B_{p-1}$ contain any arm with mean reward more than $\frac{1}{2}$, it still cannot decide whether B_p has such an arm or not. This means that if the algorithm uses too many samples in the first p passes, it risks breaking the guarantee on the sample complexity (if B_p turns out to have a high reward arm), and otherwise if it does not make enough samples, it will not gain enough "knowledge" for batch B_{p+1} and the subsequent pass.

What has changed in this argument compared to prior approaches, say, in Agarwal et al. (2022), is on how we interpreted this gain of knowledge: for us, it is quite likely that the distributions of the batches change dramatically from the original distribution after each pass; we instead *explicitly* account for the ability of the algorithm in (1) determining whether a batch contains a high reward arm, and (2) storing any high reward arm inside its memory. We shall track the probability of these events throughout the passes, sometimes even 'revealing' extra information to the algorithm that are 'not interesting', and use them inductively to establish our lower bound. This approach may be of independent interest in other settings as well that target proving multi-pass/round lower bounds on sample-space tradeoffs for learning problems.

Apart from the novel inductive argument, our techniques are distinct from Agarwal et al. (2022) on two other aspects. First, in Agarwal et al. (2022), each batch may contain the arm with reward $> \frac{1}{2}$ with *constant* probability. For the pure exploration problem, this means the best arm is among the last $\log(P)$ batches with very high probability, which makes the instance not hard. In contrast, our construction only uses $O(1/P)$ probability for each batch

to have an arm that is 'not flat' . Second, the techniques in Agarwal et al. (2022) do *not* factor in the dependence on number of arms n (namely, their bounds only hold for fixed values of n); we extend a novel 'arm-trapping' tool developed by Assadi and Wang (2022) to remedy this.

1.2. Related Work

Apart from the $O(\frac{n}{\Delta^2})$ worst-case sample complexity, pure exploration in multi-armed bandits are also studied from the lens of the *instance-sensitive* sample complexity, i.e. the bound as a function of $\{\Delta_{[i]}\}_{i=2}^{n}$, which are the mean reward gaps between the best and the i-th best arms. On this front, Karnin et al. (2013); Jamieson et al. (2014) devised algorithms that achieve $O(H_2 := \sum_{i=2}^{n} \frac{1}{\Delta_{[i]}^2} \log\log(\frac{1}{\Delta_{[i]}}))$ sample complexity, which is almost optimal up to the doubly-logarithmic term. In the streaming setting, Assadi and Wang (2022) showed that it is impossible for any algorithm with $o(n)$ memory to get the $O(H_2)$ sample complexity without strong extra conditions; on the other hand, the algorithm in Jin et al. (2021) achieves the $O(H_2)$ sample complexity in $O(\log(1/\Delta))$ passes. We note that our lower bound naturally works in the instance-sensitive setting; as such, the sharp pass-memory trade-off also applies with this sample complexity.

In addition to pure exploration, streaming MABs are studied under the context of *ε-best arm identification* and *regret minimization*. The ε-best arm identification problem aims to find an arm whose reward is at most ε-far from the best arm. On this front, the line of work by Assadi and Wang (2020); Maiti et al. (2021); Jin et al. (2021) give algorithms that finds an ε-best arm with $O(\frac{n}{\varepsilon^2})$ samples and a single arm memory. For the regret minimization task, early work of Liau et al. (2018); Chaudhuri and Kalyanakrishnan (2020) gives multi-pass streaming algorithms, and Maiti et al. (2021); Wang (2023) provided single-pass tight single-pass upper and lower regret bounds. For multi-pass scenario, Agarwal et al. (2022) provides a sharp memory-regret trade-off for multi-pass streaming MABs, and their construction shares a certain degree of similarity with ours. However, as we have discussed in Section 1.1, our techniques are substaintially different from theirs.

Aimed at modern massive data processing, MABs are also studied under other sublinear models. For instance, the settings of MABs under *collaborative learning*, in which the sampling is done by multiple agents in parallel and the goal is to minimize the rounds of communications, has been extensively studied Tao et al. (2019); Karpov et al. (2020); Karpov and Zhang (2023). We remark that the round lower bound in Tao et al. (2019) does *not* imply a lower bound in our setting: the model requires simultaneous communication and cannot be simulated by streaming algorithms efficiently. The streaming expert advice problem studied by Srinivas et al. (2022); Peng and Zhang (2023); Aamand et al. (2023) is also closely related to the streaming MABs. There, the memory complexity is defined with the classical notion of bits, which is different from the memory constraint of our model. As such, the results between the two models are not directly comparable.

2. Preliminaries

Notation. We frequently use random variables and their realizations in this paper. As a general rule, apart from a handful of self-contained proofs of technical lemmas, we use the

sans serif fonts (e.g., M) to denote the random variable and the normal font (e.g., M) to denote the realization. Throughout, we use n to denote the number of arms, μ to denote the mean rewards, and Δ to denote the (mean) reward gap between the best and the second-best best arms. As we will work on arms with Bernoulli distributions, we use $\text{Bern}(\mu)$ to denote the Bernoulli distribution with mean μ, i.e., with probability μ the realization is 1.

2.1. The Multi-pass Streaming MABs Model

We use the streaming MABs model introduced by Assadi and Wang (2020) and extended by Jin et al. (2021); Agarwal et al. (2022). Informally, the model assumes n arms arriving in a stream with an adversarial order. For each arriving arm, the algorithm is allowed to pull the arriving arm and the stored arms for an arbitrary number of times. After the arm pulls, the algorithm can (*i*). store the arriving arm; (*ii*) discard the arriving arm; and (*iii*). discard stored arms from memory. If an arm is discarded, it will not be available until its appearance in the next pass of the stream. We further assume the order of arrival is *fixed* across different passes. We define the *sample complexity* as the number of total arm pulls used by an algorithm, and the *memory complexity* as the maximum number of arms ever stored at any point in the memory. For the purpose of the lower bound proof, we allow the algorithm to store any *statistics* for free [3].

We give a formalization of the above description in what follows. We first define the deterministic algorithms before extending the notion to randomized algorithms. Let $\{\text{arm}_i\}_{i=1}^n$ be n arms with Bernoulli distributions of means $\{\mu_i\}_{i=1}^n$, i.e., the distribution for arm_i is $\text{Bern}(\mu_i)$[4]. The arms arrive one-by-one in a stream, whose order is specified by a permutation σ on $[n]$. We say ALG is a P-pass (deterministic) streaming algorithm with an s-arm memory if

- ALG maintains two objects:
  1. Memory $M \subseteq \{1, 2, \cdots, n, \perp\}^s$ and a buffer index $j_{\text{arrive}} \in [n]$ for the arriving arm. We denote M as the random variable[5] for M and $\mathbf{M}$ for the set of all possible memory M.
  2. Transcript $\Pi = ([P], [n], [n], \{0, 1\})^*$, which is an ordered list of *tuples*, and each tuple encodes the index of the pass, the index of the arriving arm (j_{arrive}), the index of the pulled arm, and the result of a single arm pull. We further denote $\mathsf{\Pi}$ as the random variable for Π and $\mathbf{\Pi}$ as the set of all possible transcripts.
- ALG has an access to a sampler $\mathcal{O} : \{\text{arm}_{\sigma(i)} \mid i \in M\} \cup \{\text{arm}_{\sigma(j_{\text{arrive}})}\} \to \{0, 1\}$ that can be repeatedly use to make a single arm pull among the stored arms and the arriving arm. After a call of $\mathcal{O}$ on the $\sigma(i)$ arm, we add tuple $(p, j_{\text{arrive}}, \sigma(i), x)$ to the transcript Π, where $x \in \{0, 1\}$ is the outcome of the arm pull.

3. Any algorithm with unbounded memory can simulate the ones with bounded statistics, and we have no rescrition on local computation power. As such, our lower bound also applies to algorithms with limited memory for statistics.
4. We work with Bernoulli distributions for a *lower bound* proof that applies to all sub-Gaussian reward distributions.
5. Although the algorithm is deterministic, there is inherent randomness from arm pulls.

- ALG has an update function $\mathcal{U} : \mathbf{M} \times [n] \times [n] \times [P] \times \mathbf{\Pi} \to \mathbf{M}$ that takes memory M, the index of the arriving arm j_{arrive}, the index of the sampled arm i, the current pass index $p \in [P]$, the past transcript Π, and the sampler $\mathcal{O}$, outputs a new memory state M.

With the above formalization, we can define the sample complexity $\mathsf{Smp}(\text{ALG})$ (total number of arm pulls) as the total number of times $\mathcal{O}$ is called, and $\mathsf{Mem}(\text{ALG}) = s$ as the maximum number of indices that can be stored (minus the one-arm buffer) at any point.

Randomized algorithms. We can extend the above notion of P-pass deterministic streaming algorithms to *randomized* algorithms in the standard manner. Concretely, a randomized algorithm with the set of internal random bits $\mathcal{R}$ can be viewed as a *distribution* over deterministic algorithms: for each $r \in \mathcal{R}$, there is a realization of a deterministic P-pass streaming algorithm. Note that similar to the storing of statistics, we do *not* charge the space for random bits, i.e., the algorithms can store an unlimited number of internal random bits for free. Since we are able to prove a lower bound under this setting, we can natrually extend the lower bound to algorithms with limited random bits.

Offline algorithms. To unify the arguments in the rest of the paper, we can define offline (i.e., classical RAM) algorithms as *simulations* of streaming algorithms under the above framework. Concretely, we can view the offline algorithm as a single-pass streaming algorithm that uses a memory of n arms. It first reads and stores all arms, and then makes calls on the sampler $\mathcal{O}$. Note also that an offline algorithm is able to simulate the *passes* and the *indices* of arms locally, i.e., to use the local memory to make an arbitrary number of (extra) passes over the stream and read an arbitrary number of arms before calling the sampler $\mathcal{O}$ with a desired j_{arrive}. As such, the transcript of an offline algorithm can be written as ordered tuples of $\Pi = (*, *, [n], \{0, 1\})^*$, where the first two elements can be modified to any index in $[P]$ and $[n]$.

Limited by space, we defer the preliminary results for single-arm sample complexity lower bounds to Appendix B.

3. Main Result

We show the formal statement of our main result in this section. We note that the formalization of Result 1 requires some work, and in particular, we need to specify the meaning of the 'lack of knowledge' on Δ by the algorithm. To this end, we define the distribution $\mathcal{D}(P, C)$ of MAB instances for any two arbitrary integers $P \geqslant 2$, $C \geqslant 1$ as follows (roughly speaking, P corresponds to the number of passes of the streaming algorithms, and C is the hidden-constant in the sample complexity of the algorithm – this will become clear shortly)[6]. An illustration of the construction of $\mathcal{D}(P, C)$ can be found in Figure 1.

Distribution $\mathcal{D}(P, C)$: A family of "hard" MAB instances for P-pass streaming algorithms.

6. We focus on $P \geqslant 2$ for technical reasons. For $P = 1$, Assadi and Wang (2022) already proved that the sample complexity is unbounded when using $o(n)$-arm memory.

1. Divide the n arms into $(P+1)$ batches $B_1, \ldots, B_{P+1}$ with equal sizes of $b := \frac{n}{P+1}$. The batches are ordered in *reverse* of the stream, i.e., in each pass, B_{P+1} arrives first, then B_P, all the way to B_1 that arrives last.
2. Initialize all the arms in every batch to have mean reward $1/2$.
3. For any batch B_p for $1 \leqslant p \leqslant P$:
   (a) Sample a coin $\Theta_p \in \{0, 1\}$ from the Bernoulli distribution $\mathrm{Bern}(\frac{1}{2P})$.
   (b) If $\Theta_p = 1$, then sample an arm uniformly at random from the batch B_p and change its mean reward to $1/2 + \eta_p$ for a parameter η_p defined as:
   $$\eta_p := \left(\frac{1}{6\,C \cdot P}\right)^{15p}. \tag{1}$$
   We refer to this arm as the **special arm** of batch B_p (which only exists if $\Theta_p = 1$).
4. For the batch B_{P+1}:
   (a) Sample an arm uniformly at random from B_{P+1} and change its mean reward to $1/2 + \eta_{P+1}$ for η_{P+1} as defined in Eq (1). We refer to this arm as the **special arm** of batch B_{P+1} (which always exists) and denote it by arm^*_{P+1}.

To continue, we need some notation. We use $I \sim \mathcal{D}(P, C)$ to denote an instance of streaming MAB sampled from the distribution $\mathcal{D}(P, C)$. For any instance I, we define $\Delta(I)$ to denote the gap between the best and second best arm. Moreover, for any instance I and integer $p \in [P+1]$, we define the following event:

- $\mathcal{E}_{\mathrm{First}}(p)$: the variables $\Theta_1 = \Theta_2 = \cdots = \Theta_{p-1} = 0$ (shorthand, $\Theta_{<p} = 0$), but $\Theta_p = 1$ (with a slight abuse of notation, we take Θ_{P+1} to be a deterministic variable which is always 1).

Notice that the events $\mathcal{E}_{\mathrm{First}}(1), \ldots, \mathcal{E}_{\mathrm{First}}(P+1)$ are mutually exclusive and exactly one of them happens for any instance. We define the **special batch** of an instance I as the batch B_p for the value of $p \in [P+1]$ where $\mathcal{E}_{\mathrm{First}}(p)$ happens.

The following observation shows that the parameter Δ of an instance I is basically determined by the choice of the special batch.

Observation 3.1 *For any $I \sim \mathcal{D}(P, C)$, if the special batch of I is B_p for $p \in [P+1]$, then*

$$\frac{1}{2} \cdot \eta_p \leqslant \Delta(I) \leqslant \eta_p.$$

Proof Note that by our construction, the best arm is the special arm of the special batch. Let p be the index of the special batch, i.e. B_p is the first batch such that $\Theta_p = 1$. We prove the upper and lower bounds separately:

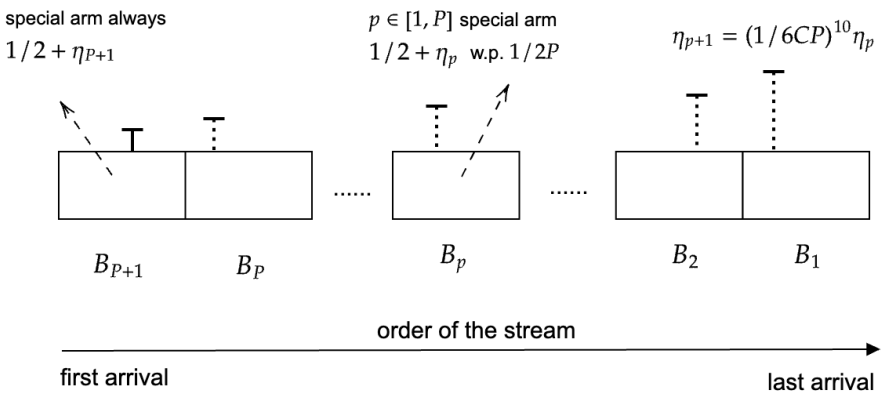

Figure 1: An illustration of $\mathcal{D}(P, C)$. The indices of batches are arranged in the *reversed* order of the arrival of the stream. Batch B_{P+1} always has an arm with $1/2+\eta_{P+1}$ mean reward, while other batches p has its special arm with mean reward $1/2+\eta_p$ with probability $\frac{1}{2P}$.

1. Upper bound: Observe that there exist (many) arms with mean reward $\frac{1}{2}$, which create a gap of η_p w.r.t. the best arm. Since $\Delta(I)$ is the *smallest* gap w.r.t. the best arm, we have $\Delta(I) \leqslant \eta_p$.

2. Lower bound: Observe that when $\Theta_p = 1$, the (potentially existing) arm with the closest mean reward is with reward $\frac{1}{2}+\eta_{p+1}$. As such, the value of $\Delta(I)$ is at least

$$\eta_p - \eta_{p+1} = \left(1 - (\frac{1}{6CP})^{15}\right) \cdot \eta_p \geqslant \frac{1}{2} \cdot \eta_p,$$

   where the last inequality is obtained by using $C \geqslant 1$ and $P \geqslant 2$.

Combining the above gives us the desired bounds. ■

By Observation 3.1, the value of Δ varies based on the realization of $\mathcal{E}_{\text{First}}(p)$ with different p values. As such, if an algorithm can achieve the optimal sample complexity bound without the knowledge of Δ given a priori, it must 'adjust' its sample complexity to be competitive with $O(n/\eta_p^2)$ *if* $\mathcal{E}_{\text{First}}(p)$ *happens*. This requirement and its consequence can be formalized in our main technical theorem as follows.

Theorem 1 *For any integers $P \geqslant 2$, $C \geqslant 1$, the following is true. Let* ALG *be any deterministic P-pass streaming algorithm that uses a memory of* $\mathsf{Mem}(\text{ALG}) \leqslant n/(20000P^3)$ *arms.*

Suppose the following is true for ALG *on instances of distribution* $\mathcal{D}(P, C)$ *and every* $p \in [P+1]$*:*

$$\mathbb{E}\left[\mathsf{Smp}(\text{ALG}) \mid \mathcal{E}_{\text{First}}(p)\right] \leqslant C \cdot \frac{n}{\eta_p^2},$$

where the randomness is taken over the choice of the instance $I \sim \mathcal{D}(P, C) \mid \mathcal{E}_{\text{First}}(p)$ *and the arm pulls. Then, the probability that* ALG *can output the best arm for* $I \sim \mathcal{D}(P, C)$ *is strictly less than* $999/1000$*.*

Theorem 1 implies that for a streaming algorithm to find the best arm with a good probability and without the a priori knowledge of Δ, it cannot simultaneous achieve i). a low number of passes, ii). a low memory, and iii). the ability to 'adjust' the sample complexity to compete with the optimal bound. As such, combining Theorem 1 with Observation 3.1 formalizes our Result 1 in the introduction.

Corollary 2 (Formalization of Result 1) *For any* $\widetilde{\Delta} > 0$*, there exists a family of streaming MABs instances* $\mathcal{D}$ *in which every instance has* $\Delta \geqslant \widetilde{\Delta}$*, such that any streaming algorithm (deterministic or randomized) that finds the best arm with an* expected *sample complexity of* $O(n/\Delta^2)$*, a success probability of at least* $1999/2000$*, and a space of* $o(n/\log^3(1/\widetilde{\Delta}))$ *arms requires* $\Omega(\frac{\log(1/\widetilde{\Delta})}{\log\log(1/\widetilde{\Delta})})$ *passes over the stream.*

Proof We first prove the statement for deterministic algorithms on the distribution $\mathcal{D}(P, C)$ with success probability $999/1000$. Our proof strategy is as follows. For any $\widetilde{\Delta}$ and any algorithm that uses $C' \cdot \frac{n}{\Delta^2(I)}$ arm pulls for arbitrary constant C', we pick appropriate P based on $\widetilde{\Delta}$ and C based on C'. Then, we sample an instance from $\mathcal{D}(P, C)$, and argue that the properties in Corollary 2 matches the properties prescribed in Theorem 1 – in particular, if the algorithm always uses $C' \cdot \frac{n}{\Delta^2(I)}$ arm pulls in expectation, the expected arm pulls of $\mathbb{E}[\mathsf{Smp}(\text{ALG}) \mid \mathcal{E}_{\text{First}}(p)]$ is at most $4C' \cdot \frac{n}{\eta_p^2}$. Finally, it turns out that the value of P is at least $\Omega\left(\frac{\log(1/\widetilde{\Delta})}{\log\log(1/\widetilde{\Delta})}\right)$ by this construction, which gives the desired lower bound.

We now formalize the above strategy. For a streaming algorithm that uses $C' \cdot \frac{n}{\Delta^2(I)}$ samples, we pick $C = 4 \cdot C'$ and use the distribution $\mathcal{D}(P, C)$ as the adversarial family of instances. We further choose $P = \Omega(\frac{\log(1/\widetilde{\Delta})}{\log\log(1/\widetilde{\Delta})})$, and observe the following properties:

- If the event $\mathcal{E}_{\text{First}}(p)$ happens in $\mathcal{D}(P, C)$, the algorithm takes at most $C \cdot \frac{n}{\eta_p^2}$ arm pulls. To see this, note that by the upper bound of the expected number of samples, and conditioning on $\mathcal{E}_{\text{First}}(p)$, there is

$$\mathbb{E}\left[\mathsf{Smp}(\text{ALG}) \mid \mathcal{E}_{\text{First}}(p)\right] \leqslant C' \cdot \frac{n}{\Delta(I)^2} = \frac{C}{4} \cdot \frac{n}{\Delta(I)^2} \leqslant \frac{C}{4} \cdot \frac{n}{(\eta_p/2)^2} = C \cdot \frac{n}{\eta_p^2},$$

  where the first inequality follows from Observation 3.1.

- For any $p \in [P+1]$, there is $\Delta \geqslant \widetilde{\Delta}$ and $\eta_p > 2 \cdot \widetilde{\Delta}$, i.e.,

$$\Delta \geqslant \frac{\eta_p}{2} \geqslant \frac{\eta_{P+1}}{2} = \frac{1}{2} \cdot \left(\frac{1}{6\,C \cdot P}\right)^{15P+15} > \widetilde{\Delta},$$

  where the last inequality is obtained by plugging in $P = \Omega\left(\frac{\log(1/\widetilde{\Delta})}{\log\log(1/\widetilde{\Delta})}\right)$.

Note that the above conditions imply (i). the expected number of samples follows the upper bound of Theorem 1; (ii). the memory follows the upper bound of Theorem 1 since $o(n/(\log(1/\widetilde{\Delta}))^3) = o(n/P^3)$ by the choice of P; and (iii). the condition of $\Delta \geqslant \widetilde{\Delta}$ is satisfied. As such, we can apply Theorem 1, and show that the algorithm must make at least $P = \Omega(\frac{\log(1/\widetilde{\Delta})}{\log\log(1/\widetilde{\Delta})}) = \Omega(\frac{\log(1/\Delta)}{\log\log(1/\Delta)})$ passes over the stream for any instance in the family, which proves the corollary for deterministic algorithms.

We now extend the result to randomized algorithms. This is a standard application of Yao's minimax principle, and we provide the proof for completeness. Assume for the purpose of contradiction that there exists a randomized algorithm with a success probability of 1999/2000 and the same restrictions as the deterministic algorithms. Let $\mathbf{R}$ be the set of internal randomness, and define $r \in \mathbf{R}$ as a *good* random string if $\Pr(\textsc{alg}$ returns the wrong arm $\mid r) \leqslant 1/1000$, where the randomness is over the inputs and the arm pulls. We say r is a *bad* random string if the above inequality does not hold. By the success probability of the algorithm, there is

$$\mathop{\mathbb{E}}_{r\in\mathbf{R}}\left[\Pr(\textsc{alg} \text{ returns the wrong arm} \mid r)\right] \leqslant \frac{1}{2000}.$$

Therefore, by the Markov bound, we have

$$\Pr(r \in \mathbf{R} \text{ is good}) \geqslant \frac{1}{2}.$$

As such, the expected sample complexity can be written as

$$\begin{aligned}\mathbb{E}\left[\mathsf{Smp}(\textsc{alg})\right] &\geqslant \mathbb{E}\left[\mathsf{Smp}(\textsc{alg}) \mid r \in \mathbf{R} \text{ is good}\right] \cdot \Pr(r \in \mathbf{R} \text{ is good}) \\ &= \frac{1}{2} \cdot \mathbb{E}\left[\mathsf{Smp}(\textsc{alg}) \mid r \in \mathbf{R} \text{ is good}\right].\end{aligned}$$

As such, by the expected sample complexity $\mathbb{E}\left[\mathsf{Smp}(\textsc{alg})\right] \leqslant O(n/\Delta^2)$ of the randomized algorithm, we have $\mathbb{E}\left[\mathsf{Smp}(\textsc{alg}) \mid r \in \mathbf{R} \text{ is good}\right] \leqslant O(n/\Delta^2)$. Therefore, for any *good* choice of r, we obtain a *deterministic* algorithm that uses $O(n/\Delta^2)$ arm pulls, a success probability of at least 999/1000, and the memory restriction of $o(n/(\log(1/\widetilde{\Delta}))^3)$ – which reaches a contradiction with the lower bound for the deterministic algorithm. ■

Note that in Corollary 2, the memory is fixed, but the sample complexity and the number of passes are allowed to be random. As long as $\widetilde{\Delta} \geqslant 2^{n^{1/3-\Omega(1)}}$, the result matches the upper bound of Jin et al. (2021) up to an exponentially smaller term.

The rest of this paper is dedicated to the proof of Theorem 1. In the next section, we state some auxiliary information-theoretic lemmas in the context of finding best arm, or rather "trapping" it, outside the streaming model. Afterward, we present the main part of our argument that uses these lemmas to establish a streaming lower bound and prove Theorem 1.

Remark 3 *We use a success probability of* $\frac{1999}{2000}$ *in Result 1 for technical convenience. For lower bounds with a lower success probability, we can apply the standard reduction argument that "boosts" the success probability. Concretely, suppose we have a* P*-pass algorithm with* s *samples,* m*-arm memory, and a success probability* $\frac{1}{2}+\varepsilon$ *for any* $\varepsilon = \Omega(1)$*. In our distribution*

$\mathcal{D}(P, C)$, we can obtain the value of Δ by the end of pass P. Therefore, we can run $O(1)$ streams in parallel, and spend $O(\frac{1}{\Delta^2})$ samples in the end to return the arm with the best empirical rewards. Such an algorithm has a success probability of $1999/2000$, an $O(m)$-arm memory, and uses $O(s)$ samples. Hence, the asymptotical sample-memory-pass trade-off remains valid for algorithms with $\frac{1}{2} + \varepsilon$ success probability for any $\varepsilon = \Omega(1)$.

4. An Overview of the Proof of Theorem 1

Additional Notation. Let $P \geqslant 2$, $C \geqslant 1$ be postitive integers, $\mathcal{D}(P, C)$ be the distribution of hard instances defined in Section 3, and ALG be a P-pass (deterministic) streaming algorithm defined as in Section 2.1. For each batch B_q, we use $I(B_q)$ to denote the *indices* in the *order of the stream* of batch q, i.e. $I(B_q) = ((P-q+1) \cdot \frac{n}{P+1}, (P-q+2) \cdot \frac{n}{P+1}]$. The batch B_q hence contains the arms $\sigma(i)$ for $i \in I(B_q)$. For any integer $p \in [P+1]$, we introduce new notation to handle variables as functions of p as follows.

- **Transcripts.** Denote $\mathsf{\Pi}^p$ and Π^p as the random variable and the realization of the transcript induced by the arm pulls *within* the p-th pass. We further define $\mathsf{\Pi}^{1:p} := (\mathsf{\Pi}^1, \ldots, \mathsf{\Pi}^p)$ and $\Pi^{1:p} := (\Pi^1, \ldots, \Pi^p)$ as the random variable and the realization of the transcript among *all* of the first p-passes. For transcripts $\mathsf{\Pi}^p$, $\mathsf{\Pi}^{1:p}$, etc., we define batch-specific transcripts as follows. We define $\mathsf{\Pi}^p_{\cap B_q}$ (resp. $\Pi^p_{\cap B_q}$) be the transcript induced by the arm pulls *on the arms in the q-th batch*, i.e. the result of $\text{arm}_{\sigma(i)}$ is recorded in $\Pi^p_{\cap B_q}$ if $i \in I(B_q)$. The notation generalizes to $\mathsf{\Pi}^{1:p}_{\cap B_q}$ as well.
- **Memory.** We use M^p and M^p to denote the random variable and the realization of the memory state by the *end* of the p-th pass.
- **Sample complexity.** We similary define $\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{q+1}}$ as the *total* number of arm pulls used on the *arms* from the $(q+1)$-th batch to the $(P+1)$-th batch, i.e. when calling the sampler $\mathcal{O}$, the index $i \in \cup_{r=q+1}^{P+1} I(B_r)$ (and it is independent of j_{arrive}). Similarly, we use $\mathsf{Smp}(\text{ALG})_{B_q}$ to denote the *total* number of arm pulls used on the arms in batch q. To avoid confusion, when we talk about the total number of arm pulls in pass p, it means the cumulative number of arm pulls in the *first p passes*.

Notice that the final output of the algorithm is a deterministic function of $(\Pi^{1:P}, M^P)$.

Since we work with the expectation over the randomness of the memory and transcript, to avoid very long lines, we sometimes use

$$\mathop{\mathbb{E}}_{\mathsf{\Pi}^{1:p-1}, \mathsf{M}^{p-1}} \left[\mathsf{Smp}(\text{ALG}) \mid \Pi^{1:p-1}, M^{p-1}, \Theta_{<p} = 0\right]$$

for the full expression of $\mathbb{E}_{\mathsf{\Pi}^{1:p-1}, \mathsf{M}^{p-1}} \left[\mathsf{Smp}(\text{ALG}) \mid \mathsf{\Pi}^{1:p-1} = \Pi^{1:p-1}, \mathsf{M}^{p-1} = M^{p-1}, \Theta_{<p} = 0\right]$. Other random variables that appear on the conditions follow the same rule for simplifications.

Memory- and Batch-obliviousness. We now introduce the notion of memory- and batch-obliviousness, which we will use crucially to describe the "limits of learning" for any streaming algorithms.

We say that the algorithm is **memory-oblivious** at the end of pass p if M^p contains no arm with reward strictly more than $1/2$ during *any* of the first p passes. Notice that

in particular if the algorithm is memory-oblivious at the end of the P-th pass, then it cannot output the best arm in the stream. We use $\mathcal{E}^p_{\text{mem-obl}}$ to denote the *event* that ALG is memory-oblivious by the *end* of pass p. Note that the memory-oblivious event has *downward implications*: if the algorithm is memory oblivious at the end of pass p, it has to be memory-oblivious for all passes $p' < p$.

We further say that the algorithm is **batch-oblivious** at the end of pass p if given $(\Pi^{1:p}, M^p)$, and conditioning on the event of $\Theta_{<p} = 0$, for any $q > p$, the algorithm "does not know" the value of Θ_q; formally,

$$\forall\ p < q \leqslant P: \quad \Pr\left(\Theta_q = 1 \mid \Pi^{1:p} = \Pi^{1:p}, \mathsf{M}^p = M^p, \Theta_{<p+1} = 0\right) \in [\frac{1}{2P} - \frac{1}{4P^2}\,,\ \frac{1}{2P} + \frac{1}{4P^2}]. \tag{2}$$

We use $\mathcal{E}^p_{\text{batch-obl}}$ to denote the *event* that ALG is batch-oblivous by the *end* of pass p.

The strategy of the proof. We will inductively show that the algorithm is going to be memory-oblivious and batch-oblivious with a large probability throughout each pass. To do so, we consider two types of possible behavior for the algorithm ALG in each pass p:

- **Conservative** case: the first case is when the algorithm decides to be "conservative" with its arm pulls in the first p passes. We show that if the probability for the algorithm to be in conservative case after the first $p-1$ passes is large, and the algorithm decides to be conservative on the first p passes, then the algorithm is going to remain memory-oblivious and batch-oblivious for the subsequent pass as well with a sufficiently large probability.
- **Radical** case: the complementary case is when the algorithm decides to make "many" arm pulls in the first p passes. In this case, we use the memory- and batch-obliviousness properties of the algorithm to show that such an algorithm is necessarily going to break the guarantee on the number of arm pulls imposed on it by Theorem 1 in some cases.

Formalizing this strategy is challenging due to the nature of the guarantee of Theorem 1 on the event $\mathcal{E}_{\text{First}}(p)$ for some unknown p (rather informally speaking, since η_p is unknown to the algorithm, but also follows a certain distribution in the input). This requires a careful conditioning on various events happening in the algorithm and keeping track of the information revealed by these events. Limited by space, we only present the main lemmas of both cases, and use them to prove the main lower bound. We defer the full analysis to Appendix D.

The main lemma for the conservative case. Our main lemma for the conservative case is as follows.

Lemma 4 (Conservative case) *For any integer $p \in [P]$, let* ALG *be a streaming algorithm with a memory of at most $n/(20000P^3)$ arms, and assume that at the end of the pass $p-1$, the following conditions hold*

(I). The probability for $\mathcal{E}^{p-1}_{batch\text{-}obl}, \mathcal{E}^{p-1}_{mem\text{-}obl}$ and $\Theta_{<p} = 0$ to happen is large, i.e.,

$$\Pr\left(\mathcal{E}^{p-1}_{batch\text{-}obl}, \mathcal{E}^{p-1}_{mem\text{-}obl}, \Theta_{<p} = 0\right) \geqslant \left(1 - \frac{1}{2P}\right)^{10(p-1)};$$

(II). Conditioning on $\Theta_{<p} = 0$ *and* $\mathcal{E}^{p-1}_{\textit{batch-obl}}, \mathcal{E}^{p-1}_{\textit{mem-obl}}$*, the expected number of arm pulls (over the randomness of the first* p *passes) is small, i.e.,*

$$\mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \mathcal{E}^{p-1}_{\textit{batch-obl}}, \mathcal{E}^{p-1}_{\textit{mem-obl}}, \Theta_{<p} = 0\right] \leqslant \frac{1}{10^9} \cdot \frac{n}{\gamma_{p+1} \cdot P^{30}}.$$

Then, with probability at least $(1 - 1/2P)^{10p}$*, at the end of pass* p*, we have* $\Theta_{<p+1} = 0$ *and the algorithm is memory- and batch-oblivious, i.e.,*

$$\Pr\left(\mathcal{E}^{p}_{\textit{batch-obl}}, \mathcal{E}^{p}_{\textit{mem-obl}}, \Theta_{<p+1} = 0\right) \geqslant \left(1 - \frac{1}{2P}\right)^{10p}.$$

The main lemma for the radical case. In contrast to the conservative case, our main lemma for the radical case is as follows.

Lemma 5 (Radical case) *For any integer* $p \in [P]$*, suppose a streaming algorithm* ALG *is memory- and batch-oblivious at the end of the pass* $p-1$*, and that the underlying instance satisfies* $\Theta_{<p} = 0$*. Additionally, suppose*

$$\mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \mathcal{E}^{p-1}_{\textit{batch-obl}}, \mathcal{E}^{p-1}_{\textit{mem-obl}}, \Theta_{<p} = 0\right] > \frac{1}{10^9} \cdot \frac{n}{\gamma_{p+1} \cdot P^{30}};$$

then,

$$\mathbb{E}\left[\mathsf{Smp}(\text{ALG}) \mid \mathcal{E}^{p-1}_{\textit{batch-obl}}, \mathcal{E}^{p-1}_{\textit{mem-obl}}, \mathcal{E}_{\text{First}}(p)\right] > 20000 \cdot C \cdot \frac{n}{\eta_p^2}.$$

Putting Everything Together: Proof of Theorem 1. We now prove Theorem 1 with Lemmas 4 and 5. We remind the readers that we use ALG to denote the streaming algorithm. Note that in the beginning of the first pass, ALG is necessarily memory- and batch-oblivious since there is $\Pi^0 = \emptyset$ and $M^0 = \emptyset$. Therefore, by Lemma 5, if the algorithm enters the *radical case*, there is

$$\mathbb{E}\left[\mathsf{Smp}(\text{ALG}) \mid \mathcal{E}_{\text{First}}(1)\right] = \mathbb{E}\left[\mathsf{Smp}(\text{ALG}) \mid \Theta_1 = 1\right] > C \cdot \frac{n}{\eta_1^2},$$

which breaks the sample complexity requirement in Theorem 1. Therefore, ALG must use the *conservative case* for the first pass.

Starting from the second pass, we argue that no pass should use the *radical case* if ALG is to follow the upper bound on the sample complexity as required by Theorem 1. Suppose $\tilde{p}$ is the first pass that the algorithm enters the *radical case*, and since we have the base case of $p = 1$ and the condition of Lemma 4 (conservative case) being satisfied before pass $\tilde{p}$, there is

$$\Pr\left(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}}, \Theta_{<\tilde{p}} = 0\right) \geqslant \left(1 - \frac{1}{2P}\right)^{10(\tilde{p}-1)}.$$

We use the above result to lower bound the probability for $\Pr(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}} \mid \mathcal{E}_{\text{First}}(\tilde{p}))$, which will eventually lead to a lower bound on $\mathbb{E}\left[\mathsf{Smp}(\text{ALG}) \mid \mathcal{E}_{\text{First}}(\tilde{p})\right]$ that breaks the limit of samples.

To this end, we first show the following technical claim that allows us to "drop" conditions on $\Theta_{\tilde{p}}$ conditioning on $\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}$ and $\mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}}$. Intuitively, such a claim is true by the obliviousness of the transcipt on $\Theta_{\tilde{p}}$, which is similar-in-spirit with Lemma 18.

Claim 4.1 *The following statement is true:*

$$\Pr\left(\mathcal{E}^{\tilde{p}-1}_{\textit{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\textit{mem-obl}} \mid \mathcal{E}_{\text{First}}(\tilde{p})\right) \geqslant \frac{1}{2} \cdot \Pr\left(\mathcal{E}^{\tilde{p}-1}_{\textit{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\textit{mem-obl}}, \Theta_{<\tilde{p}} = 0\right).$$

The proof of Claim 4.1 can be found in the full analysis of Appendix D. We now establish the lower bound on the expected sample for pass $\tilde{p}$. By Claim 4.1, we have that

$$\Pr\left(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}} \mid \mathcal{E}_{\text{First}}(\tilde{p})\right) \geqslant \frac{1}{2} \cdot \left(1 - \frac{1}{2P}\right)^{10(\tilde{p}-1)} > \frac{1}{1000}, \tag{3}$$

where the first inequality uses Claim 4.1 and the lower bound on $\Pr(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}}, \Theta_{<\tilde{p}} = 0)$, and the last inequality is obtained by using $\left(1 - \frac{1}{2P}\right)^{10P-10} > \frac{1}{500}$ for any $P \geqslant 2$. Therefore, we can bound the sample complexity of the algorithm if it enters the *radical case* on the $\tilde{p}$-th pass as follows.

$$\begin{aligned}
&\mathbb{E}\left[\mathsf{Smp}(\text{ALG}) \mid \mathcal{E}_{\text{First}}(\tilde{p})\right] \\
&\geqslant \mathbb{E}\left[\mathsf{Smp}(\text{ALG}) \mid \mathcal{E}_{\text{First}}(\tilde{p}), \mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}}\right] \cdot \Pr\left(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}} \mid \mathcal{E}_{\text{First}}(\tilde{p})\right) \\
&> 20000C \cdot \frac{n}{\eta_p^2} \cdot \frac{1}{1000} > C \cdot \frac{n}{\eta_p^2},
\end{aligned}$$

which breaks the requirement of sample complexity bound in Theorem 1. As such, to keep the promise on the sample complexity, ALG has to be in the *conservative case* for *all* P passes.

Now, we can apply the calculation in Eq (3) again to argue that with probability strictly more than $\frac{1}{1000}$, after the P-th pass, we obtain transcript and memory that are memory- and batch-oblivious. As such, no arm with a mean reward strictly more than $1/2$ will be in the memory of ALG, which means the success probability is strictly less than $\frac{999}{1000}$.

Acknowledgments

We thank anonymous COLT reviewers for helpful comments. Part of this work was done while both authors were at Rutgers University and were supported in part by an NSF CAREER Grant CCF-2047061, a gift from Google Research, and a Fulcrum award from Rutgers Research Council.

References

Anders Aamand, Justin Y. Chen, Huy Lê Nguyen, and Sandeep Silwal. Improved space bounds for learning with experts. *CoRR*, abs/2303.01453, 2023. doi: 10.48550/arXiv.2303.01453. URL https://doi.org/10.48550/arXiv.2303.01453.

Arpit Agarwal, Shivani Agarwal, Sepehr Assadi, and Sanjeev Khanna. Learning with limited rounds of adaptivity: Coin tossing, multi-armed bandits, and ranking from pairwise comparisons. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017*, pages 39–75, 2017.

Arpit Agarwal, Sanjeev Khanna, and Prathamesh Patil. A sharp memory-regret trade-off for multi-pass streaming bandits. In Po-Ling Loh and Maxim Raginsky, editors, *Conference on Learning Theory, 2-5 July 2022, London, UK*, volume 178 of *Proceedings of Machine Learning Research*, pages 1423–1462. PMLR, 2022. URL https://proceedings.mlr.press/v178/agarwal22a.html.

Deepak Agarwal, Bee-Chung Chen, and Pradheep Elango. Explore/exploit schemes for web content optimization. In Wei Wang, Hillol Kargupta, Sanjay Ranka, Philip S. Yu, and Xindong Wu, editors, *ICDM 2009, The Ninth IEEE International Conference on Data Mining, Miami, Florida, USA, 6-9 December 2009*, pages 1–10. IEEE Computer Society, 2009. doi: 10.1109/ICDM.2009.52. URL https://doi.org/10.1109/ICDM.2009.52.

Sepehr Assadi and Chen Wang. Exploration with limited memory: streaming algorithms for coin tossing, noisy comparisons, and multi-armed bandits. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1237–1250. ACM, 2020. doi: 10.1145/3357713.3384341. URL https://doi.org/10.1145/3357713.3384341.

Sepehr Assadi and Chen Wang. Single-pass streaming lower bounds for multi-armed bandits exploration with instance-sensitive sample complexity. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022 (to appear)*, 2022.

Maryam Aziz, Emilie Kaufmann, and Marie-Karelle Riviere. On multi-armed bandit designs for dose-finding clinical trials. *The Journal of Machine Learning Research*, 22(1):686–723, 2021.

Dimitris Bertsimas and Adam J. Mersereau. A learning approach for interactive marketing to a customer segment. *Oper. Res.*, 55(6):1120–1135, 2007. doi: 10.1287/opre.1070.0427. URL https://doi.org/10.1287/opre.1070.0427.

Arghya Roy Chaudhuri and Shivaram Kalyanakrishnan. Regret minimisation in multi-armed bandits using bounded arm memory. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 10085–10092. AAAI Press, 2020. URL https://ojs.aaai.org/index.php/AAAI/article/view/6566.

Lijie Chen, Jian Li, and Mingda Qiao. Nearly instance optimal sample complexity bounds for top-k arm selection. In Aarti Singh and Xiaojin (Jerry) Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 101–110. PMLR, 2017. URL http://proceedings.mlr.press/v54/chen17a.html.

Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006. ISBN 978-0-471-24195-9.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. PAC Bounds for Multi-Armed Bandit and Markov Decision Processes. In *COLT*, 2002.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7:1079–1105, 2006.

Kevin G. Jamieson, Matthew Malloy, Robert D. Nowak, and Sébastien Bubeck. lil' UCB : An optimal exploration algorithm for multi-armed bandits. In Maria-Florina Balcan, Vitaly Feldman, and Csaba Szepesvári, editors, *Proceedings of The 27th Conference on Learning Theory, COLT 2014, Barcelona, Spain, June 13-15, 2014*, volume 35 of *JMLR Workshop and Conference Proceedings*, pages 423–439. JMLR.org, 2014. URL http://proceedings.mlr.press/v35/jamieson14.html.

Tianyuan Jin, Keke Huang, Jing Tang, and Xiaokui Xiao. Optimal streaming algorithms for multi-armed bandits. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5045–5054. PMLR, 2021. URL http://proceedings.mlr.press/v139/jin21a.html.

Shivaram Kalyanakrishnan and Peter Stone. Efficient Selection of Multiple Bandit Arms: Theory and Practice. In *ICML*, 2010.

Zohar Shay Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 1238–1246. JMLR.org, 2013. URL http://proceedings.mlr.press/v28/karnin13.html.

Nikolai Karpov and Qin Zhang. Communication-efficient collaborative best arm identification. In *Proc. AAAI Conference on Artificial Intelligence (AAAI 23)*, 2023.

Nikolai Karpov, Qin Zhang, and Yuan Zhou. Collaborative top distribution identifications with limited interaction (extended abstract). In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 160–171. IEEE, 2020. doi: 10.1109/FOCS46700.2020.00024. URL https://doi.org/10.1109/FOCS46700.2020.00024.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti, editors, *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 661–670. ACM, 2010. doi: 10.1145/1772690.1772758. URL https://doi.org/10.1145/1772690.1772758.

Shaoang Li, Lan Zhang, Junhao Wang, and Xiang-Yang Li. Tight memory-regret lower bounds for streaming bandits. *CoRR*, abs/2306.07903, 2023. doi: 10.48550/arXiv.2306.07903. URL https://doi.org/10.48550/arXiv.2306.07903.

David Liau, Zhao Song, Eric Price, and Ger Yang. Stochastic multi-armed bandits in constant space. In Amos J. Storkey and Fernando Pérez-Cruz, editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, volume 84 of *Proceedings of Machine Learning Research*, pages 386–394. PMLR, 2018. URL http://proceedings.mlr.press/v84/liau18a.html.

Arnab Maiti, Vishakha Patil, and Arindam Khan. Multi-armed bandits with bounded arm-memory: Near-optimal guarantees for best-arm identification and regret minimization. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 19553–19565, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/a2f04745390fd6897d09772b2cd1f581-Abstract.html.

Shie Mannor and John N Tsitsiklis. The Sample Complexity of Exploration in the Multi-Armed Bandit Problem. *Journal of Machine Learning Research*, 5:623–648, 2004.

Binghui Peng and Fred Zhang. Online prediction in sub-linear space. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 1611–1634. SIAM, 2023. doi: 10.1137/1.9781611977554.ch60. URL https://doi.org/10.1137/1.9781611977554.ch60.

Eric M. Schwartz, Eric T. Bradlow, and Peter S. Fader. Customer acquisition via display advertising using multi-armed bandit experiments. *Mark. Sci.*, 36(4):500–522, 2017. doi: 10.1287/mksc.2016.1023. URL https://doi.org/10.1287/mksc.2016.1023.

Vaidehi Srinivas, David P. Woodruff, Ziyu Xu, and Samson Zhou. Memory bounds for the experts problem. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1158–1171. ACM, 2022. doi: 10.1145/3519935.3520069. URL https://doi.org/10.1145/3519935.3520069.

Chao Tao, Qin Zhang, and Yuan Zhou. Collaborative learning with limited interaction: Tight bounds for distributed exploration in multi-armed bandits. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 126–146. IEEE Computer Society, 2019. doi: 10.1109/FOCS.2019.00017. URL https://doi.org/10.1109/FOCS.2019.00017.

Sofía S Villar, Jack Bowden, and James Wason. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 30(2):199, 2015.

Chen Wang. Tight regret bounds for single-pass streaming multi-armed bandits. In *Proceedings of the 40th International Conference on Machine Learning, ICML 2023 (To appear)*, Proceedings of Machine Learning Research, 2023.

Appendix A. Standard Technical Tools

A.1. Statistical Distances

We introduce the widely-used statistical distance notions of Kullback–Leibler divergence (KL divergence) and total variation distance (TVD) in this section.

KL divergence and its properties. We start with introducing the Kullback–Leibler divergence (KL divergence) and its properties.

Definition 6 (KL divergence) *Let X and Y be two discrete random variables supported over the same Ω, and let their distributions be μ_X and μ_Y. The KL divergence between X and Y, denoted as $\mathbb{D}(X \mid\mid Y)$, is defined as*

$$\mathbb{D}(X \mid\mid Y) = \sum_{\omega \in \Omega} \mu_X(\omega) \log \left(\frac{\mu_X(\omega)}{\mu_Y(\omega)} \right).$$

Total variation distance and its properties. Similar to the KL divergence we defined above, the total variation distance (TVD) is another statistical distance between two distributions.

Definition 7 *Let X and Y be two random variables supported over the same Ω, and let μ_X and μ_Y be their probability measures. The total variation distance (TVD) is between X and Y is defined as*

$$\|X - Y\|_{\text{tvd}} = \sup_{\Omega' \subseteq \Omega} \left| \mu_X(\Omega') - \mu_Y(\Omega') \right|.$$

In particular, when the random variables are discrete, we have

$$\|X - Y\|_{\text{tvd}} = \frac{1}{2} \sum_{\omega \in \Omega} \left| \mu_X(\omega) - \mu_Y(\omega) \right|.$$

A.2. Statistical Distances and Their Properties

We shall use the following standard properties of KL-divergence and TVD defined in Section A.1. For the proof of this results, see the excellent textbook by Cover and Thomas Cover and Thomas (2006).

The following facts state the chain rule property and convexity of KL-divergence.

Fact A.1 (Chain rule of KL divergence) *For any random variables $X = (X_1, X_2)$ and $Y = (Y_1, Y_2)$ be two random variables,*

$$\mathbb{D}(X \mid\mid Y) = \mathbb{D}(X_1 \mid\mid Y_1) + \mathop{\mathbb{E}}_{x \sim X_1} \mathbb{D}(X_2 \mid X_1 = x \mid\mid Y_2 \mid Y_1 = x).$$

Fact A.2 (Convexity KL-divergence) *For any distributions μ_1, μ_2 and ν_1, ν_2 and any $\lambda \in (0, 1)$,*

$$\mathbb{D}(\lambda \cdot \mu_1 + (1-\lambda) \cdot \mu_2 \mid\mid \lambda \cdot \nu_1 + (1-\lambda) \cdot \nu_2) \leqslant \lambda \cdot \mathbb{D}(\mu_1 \mid\mid \nu_1) + (1-\lambda) \cdot \mathbb{D}(\mu_2 \mid\mid \nu_2).$$

Fact A.3 (Conditioning cannot decrease KL-divergence) *For any random variables X, Y, Z,*

$$\mathbb{D}(X \mid\mid Y) \leqslant \mathop{\mathbb{E}}_{z\sim Z} \mathbb{D}(X \mid Z = z \mid\mid Y \mid Z = z).$$

Pinsker's inequality relates KL-divergence to TVD.

Fact A.4 (Pinsker's inequality) *For any random variables X and Y supported over the same Ω,*

$$\|X - Y\|_{\text{tvd}} \leqslant \sqrt{\frac{1}{2} \cdot \mathbb{D}(X \mid\mid Y)}.$$

The following fact characterizes the error of MLE for the source of a sample based on the TVD of the originating distributions.

Fact A.5 *Suppose μ and ν are two distributions over the same support Ω; then, given one sample s from the following distribution*

- *With probability ρ, sample s from μ;*
- *With probability $1 - \rho$, sample s from ν;*

The best probability we can decide whether s came from μ or ν is

$$\max(\rho, 1-\rho) + \min(\rho, 1-\rho) \cdot \|\mu - \nu\|_{\text{tvd}}.$$

A.3. Information Theory Tools

We use information-theoretic tools in our proofs, and we include a review of the basic notions and properties used therein. We start with the definition of entropy. For a random variable X, we let $\mathbb{H}(X)$ be the *Shannon entropy* of X, defined as follows

Definition 8 *Let X be a discrete random variable with distributions μ_X, the* Shannon entropy *of X is defined as*

$$\mathbb{H}(X) := \mathbb{E}\left[\log(1/\mu(X))\right] = \sum_{x \in supp(X)} \mu(x) \cdot \log(\frac{1}{\mu(x)}),$$

where $supp(X)$ is the support of X. If X is a Bernoulli random variable, we use $H_2(p)$ to denote its Shannon entropy, where P is the probability for $X = 1$.

We now give the definition of conditional entropy and mutual information.

Definition 9 *Let X, Y be two random variables, we define the* conditional entropy *as*

$$\mathbb{H}(X|Y) = \mathbb{E}_{y\sim Y}[\mathbb{H}(X \mid Y = y)].$$

With conditional entropy, we can define the mutual information *between X and Y as*

$$\mathbb{I}(X;Y) := \mathbb{H}(X) - \mathbb{H}(X \mid Y) = \mathbb{H}(Y) - \mathbb{H}(Y|X).$$

The following information-theoretic facts (see e.g. Cover and Thomas (2006) for details) are used in our lower bound proofs.

Fact A.6 *Let X, Y, Z be three discrete random variables:*

- *KL-divergence view of mutual information:* $\mathbb{I}(X;Y) = \mathbb{E}_{y\sim Y}\left[\mathbb{D}(X \mid Y = y \mid\mid X)\right]$.
- $0 \leqslant \mathbb{H}(X) \leqslant \log(|supp(X)|)$. *In particular, if X is a Bernoulli random variable, there is* $H_2(p) \leqslant 1$.
- $0 \leqslant \mathbb{I}(X;Y) \leqslant \min\{\mathbb{H}(X), \mathbb{H}(Y)\}$.
- *Conditioning on independent random variable: let X be independent of Z, then* $\mathbb{I}(X;Y) \leqslant \mathbb{I}(X;Y \mid Z)$.
- *Chain rule of mutual information:* $\mathbb{I}(X,Y;Z) = \mathbb{I}(X;Z) + \mathbb{I}(Y;Z \mid X)$.
- *Sub-additivity of entropy:* $\mathbb{H}(X,Y) \leqslant \mathbb{H}(X) + \mathbb{H}(Y)$, *where $\mathbb{H}(X,Y)$ is the joint entropy of variables X, Y.*
- *Conditional independence of entropy:* $\mathbb{H}(X \mid Y,Z) = \mathbb{H}(X \mid Y)$ *if $X \perp Z \mid Y$, where the $\perp$ notation stands for independence.*

The following statement is known as the *data processing inequality*, which says if Y is obtained as a function of X, and Z is obtained as a function of Y, then the mutual information between X and Z can only be lower than that between X and Y.

Proposition 10 *Let X, Y, and Z be random variables on finite supports, and we slightly abuse the notation to let X, Y, Z to denote the distribution functions as well. Let f be a deterministic function (no internal randomness), and suppose $Z = f(Y)$. Then, we have*

$$\mathbb{I}(X;Z) \leqslant \mathbb{I}(X;Y).$$

The following statement characterizes the relationship between the "zero mutual information" and the independence of the conditional probability.

Proposition 11 *Let X, Y, and Z be random variables on finite supports, and suppose* $\mathbb{I}(X;Y \mid Z = z) = 0$. *Then, for any realization $y \in Y$, there is*

$$\Pr(X \mid Z = z, Y = y) = \Pr(X \mid Z = z).$$

Appendix B. Standard Sample Complexity Lower Bounds for Single-armed Bandit

We present lower bounds on the necessary number of arm pulls to identify the reward of a *single* arm. These lower bounds serve as the basis for the reduction proofs we used in the auxiliary lemmas (Lemma 14 and Lemma 15). We remark that the lemmas are not limited to the streaming setting and they hold even for classical algorithms.

Our first lemma shows that to *determine* the mean reward of an arm from distributions with gap β, an $\Omega(1/\beta^2)$ number of arm pulls is necessary.

Lemma 12 *Consider an arm with a Bernoulli distribution whose mean is parameterized as follows.*

- *With probability ρ, the mean reward is $\frac{1}{2}+\beta$;*
- *With probability $1-\rho$, the mean reward is $\frac{1}{2}$;*

where $\rho \in (0, \frac{1}{2}]$ is a fixed parameter. Any algorithm to determine the reward of the arm for $\beta \in (0, \frac{1}{6})$ and a success probability of at least $(1-\rho+\varepsilon)$ has to use $\frac{1}{4}\cdot\frac{\varepsilon^2}{\rho^2\beta^2}$ arm pulls.

Proof Let X be the random variable for the Bernoulli distribution with mean $\frac{1}{2}+\beta$ and Y the random variable for the Bernoulli distribution with mean $\frac{1}{2}$. We use Fact A.5 to argue that for a single arm pull, the probability for the algorithm to not identify the correct case is at least $\rho\cdot(1-\|X-Y\|_{\text{tvd}})$. On the other hand, note that for the two Bernoulli distributions with means $\frac{1}{2}+\beta$ and $\frac{1}{2}$, there KL-divergence can be bounded as

$$\begin{aligned}
\mathbb{D}(X \mid\mid Y) &= (\frac{1}{2}+\beta)\cdot\log(1+2\beta) + (\frac{1}{2}-\beta)\cdot\log(1-2\beta) \\
&= \frac{1}{2}\cdot\log((1+2\beta)(1-2\beta)) + \beta\cdot\log(\frac{1+2\beta}{1-2\beta}) \\
&\leqslant \beta\cdot\log(\frac{1+2\beta}{1-2\beta}) && (\log(1-4\beta^2)<0) \\
&\leqslant \beta\cdot\log(2^{6\cdot\beta}) && (\tfrac{1+2\beta}{1-2\beta}\leqslant 2^{6\cdot\beta} \text{ for } \beta\in(0,\tfrac{1}{6})) \\
&= 6\cdot\beta^2.
\end{aligned}$$

As such, using Pinsker's inequality (Fact A.4) that $\|X-Y\|_{\text{tvd}} \leqslant \sqrt{\frac{1}{2}\cdot\mathbb{D}(X \mid\mid Y)}$, and obtain that the probability for the algorithm to incorrectly identify the arm is at least $\rho\cdot(1-\sqrt{\frac{1}{2}\cdot\mathbb{D}(X \mid\mid Y)})$. The bound can be generalized to s samples: let $X^{[s]}$ and $Y^{[s]}$ be the distributions of s samples from X and Y. Then, we have:

$$\Pr\left(\text{algorithm makes wrong prediction}\right) \geqslant \rho\cdot\left(1-\sqrt{\frac{1}{2}\cdot\mathbb{D}(X^{[s]} \mid\mid Y^{[s]})}\right).$$

Using the fact that the samples are from independent and identical random variables, we can factorize $X^{[s]}$ with the marginal random variables of $\{X^i\}_{i=1}^s$ by the chain rule as follows:

$$\begin{aligned}
\mathbb{D}(X^{[s]} \mid\mid Y^{[s]})) &= \mathbb{D}(X^s \mid\mid Y^s) + \mathbb{D}(X^{[s-1]} \mid X^s \mid\mid Y^{[s-1]} \mid Y^s) && \text{(by the chain rule)} \\
&= \mathbb{D}(X \mid\mid Y) + \mathbb{D}(X^{[s-1]} \mid\mid Y^{[s-1]}) && \text{(i.i.d. random variables)} \\
&= \cdots\cdots\cdots \\
&= s\cdot\mathbb{D}(X \mid\mid Y).
\end{aligned}$$

Therefore, combining the above steps, we have

$$\begin{aligned}\Pr\left(\text{algorithm makes wrong prediction}\right) &\geqslant \rho \cdot \left(1 - \sqrt{\frac{1}{2} \cdot \mathbb{D}(X^s \mid\mid y^s)}\right)\\ &\geqslant \rho \cdot \left(1 - \sqrt{\frac{1}{2} \cdot 6s \cdot \beta^2}\right)\\ &\geqslant \rho \cdot \left(1 - 2 \cdot \beta \cdot \sqrt{s}\right).\end{aligned}$$

On the other hand, we want the error probability to be at most $\rho - \varepsilon$, which means $\rho \cdot \left(1 - 2\beta \cdot \sqrt{s}\right) \leqslant \rho - \varepsilon$, which solves to $s \geqslant \frac{1}{4} \cdot \frac{\varepsilon^4}{\rho^2\beta^2}$.

■

We further provide a lemma showing that if the number of arm pulls is small, the "knowledge" of the algorithm cannot change the distribution for which case the instance is from by too much. More formally, we prove that with a limited number of arm pulls, from the algorithm's perspective, the probability for which case the instance is from remains close to the original distribution.

Lemma 13 *Let $\beta \in (0, \frac{1}{6})$ and $\rho \in (0, \frac{1}{2})$. Sample Θ from $\{0, 1\}$ such that $\Theta = 1$ with probability ρ. Consider an arm with a Bernoulli distribution from the following family:*

- *If $\Theta = 1$, the distribution is $\text{Bern}(1/2 + \beta)$;*
- *If $\Theta = 0$, the distribution is $\text{Bern}(1/2)$;*

Let ALG be an algorithm that uses at most $s = \frac{1}{12} \cdot \frac{\varepsilon^3}{\rho \cdot \beta^2}$ arm pulls on an instance I sampled from the family. Let Π be the transcript of ALG, and let $\mathsf{\Pi}$ be the random variable of Π. Then, with probability at least $1 - \varepsilon$ over the randomness of transcript $\mathsf{\Pi}$, there is

$$\begin{aligned}&\Pr\left(\Theta = 1 \mid \mathsf{\Pi} = \Pi\right) \in [\rho - \varepsilon, \rho + \varepsilon];\\ &\Pr\left(\Theta = 0 \mid \mathsf{\Pi} = \Pi\right) \in [1 - \rho - \varepsilon, 1 - \rho + \varepsilon].\end{aligned}$$

Proof Recall that $\mathsf{\Pi}$ is the random variable for the transcript of the algorithm, and Θ is the random variable that controls from which case the instance is sampled. We can write $\mathsf{\Pi} = (\mathsf{\Pi}_1, \mathsf{\Pi}_2, \ldots, \mathsf{\Pi}_s)$, where $\mathsf{\Pi}_i$ denotes the random variable for the tuple of the i-th armed pull (recall that $\mathsf{\Pi}$ and its realization Π are defined as ordered tuples in Section 2.1). We

have,

$$\begin{aligned}
\mathbb{I}(\Theta\,;\Pi) &= \sum_{i=1}^{s} \mathbb{I}(\Theta\,;\Pi_i \mid \Pi^{<i}) && \text{(by chain rule of mutual information)}\\
&= \sum_{i=1}^{s} \mathbb{H}(\Pi_i \mid \Pi^{<i}) - \mathbb{H}(\Pi_i \mid \Theta, \Pi^{<i}) && \text{(by the definition of mutual information)}\\
&\leqslant \sum_{i=1}^{s} \mathbb{H}(\Pi_i) - \mathbb{H}(\Pi_i \mid \Theta, \Pi^{<i}) && \text{(conditioning can only reduce the entropy)}\\
&= \sum_{i=1}^{s} \mathbb{H}(\Pi_i) - \mathbb{H}(\Pi_i \mid \Theta)\\
&\text{(because } \Pi_i \perp \Pi^{<i} \mid \Theta \text{ as knowing } \Theta \text{ fixes distribution of } \Pi_i \text{ to be either Bern}(1/2+\beta) \text{ or Bern}(1/2))\\
&= \sum_{i=1}^{s} \mathbb{I}(\Theta\,;\Pi_i) && \text{(by the definition of mutual information)}\\
&= \sum_{i=1}^{s} \mathop{\mathbb{E}}_{\theta\in\{0,1\}} [\mathbb{D}(\Pi_i \mid \Theta = \theta \mid\mid \Pi_i)]\\
&&& \text{(by the connection of KL-divergence with mutual information)}\\
&= \sum_{i=1}^{s} \rho\cdot\mathbb{D}(\Pi_i \mid \Theta = 1 \mid\mid \Pi_i) + (1-\rho)\cdot\mathbb{D}(\Pi_i \mid \Theta = 0 \mid\mid \Pi_i)\\
&&& \text{(by the distribution of } \theta)\\
&= \sum_{i=1}^{s} \rho\cdot\mathbb{D}(\text{Bern}(\frac{1}{2}+\beta) \mid\mid \text{Bern}(\frac{1}{2}+\rho\cdot\beta)) + (1-\rho)\cdot\mathbb{D}(\text{Bern}(\frac{1}{2}) \mid\mid \text{Bern}(\frac{1}{2}+\rho\cdot\beta))\\
&&& \text{(as distribution of } \Pi_i \text{ is } \rho\cdot\text{Bern}(\tfrac{1}{2}+\beta) + (1-\rho)\cdot\text{Bern}(\tfrac{1}{2}) = \text{Bern}(\tfrac{1}{2}+\rho\cdot\beta))\\
&\leqslant s\cdot\left(\rho\cdot 6\cdot(\rho\cdot\beta-\beta)^2 + 6\cdot(1-\rho)\cdot(\rho\cdot\beta)^2\right) && \text{(as proven in Lemma 12)}\\
&\leqslant s\cdot\left(12\rho\cdot\beta^2\right) \leqslant \varepsilon^3. && \text{(by the upper bound on } s)
\end{aligned}$$

The above calculation also implies that

$$\mathbb{I}(\Theta\,;\Pi) = \mathop{\mathbb{E}}_{\Pi}[\mathbb{D}(\Theta \mid\mid \Theta \mid \Pi = \Pi)] \leqslant \varepsilon^3.$$

By Markov bound, with probability $1-\varepsilon$ over the choice of $\Pi\sim\Pi$, we have

$$\mathbb{D}(\Theta \mid\mid \Theta \mid \Pi = \Pi) \leqslant \varepsilon^2.$$

By Pinsker's inequality (Fact A.4), for any such Π, we have,

$$\|\Theta - \Theta \mid \Pi = \Pi\|_{\text{tvd}} \leqslant \varepsilon.$$

By the definition of total variation distance, this implies that

$$|\Pr(\Theta = 0) - \Pr(\Theta = 0 \mid \Pi = \Pi)| + |\Pr(\Theta = 1) - \Pr(\Theta = 1 \mid \Pi = \Pi)| \leqslant \varepsilon.$$

By upper bounding each term separately and using the distribution of Θ, we have,

$$|\Pr(\Theta = 0 \mid \Pi = \Pi) - (1-\rho)| \leqslant \varepsilon \quad \text{and} \quad |\Pr(\Theta = 1 \mid \Pi = \Pi) - \rho| \leqslant \varepsilon,$$

which concludes the proof. ■

Appendix C. Auxiliary Lemmas for Pure Exploration in MABs

We present two auxiliary lemmas in this section that are needed for our main proof. These lemmas concern MABs for *offline* algorithms, i.e., without any streaming restriction, and they can be used in the streaming setting with arbitrary pass and j_{arrive} index (see Section 2.1 for the detailed discussion). The proofs are rather standard application of known ideas. However, we are not aware of an exact formulation of these lemmas in prior work that we need in our main proofs in the subsequent section; thus, for completeness, we present and prove these lemmas in this section.

The first lemma is a generalization of the arm-trapping lemma of Assadi and Wang (2022) to the case when success probability can be quite small.

Lemma 14 (low-probability arm-trapping lemma) *Suppose we have a set of $k \geqslant 1$ arms with mean reward $1/2$ and we pick one of them uniformly at random – called the **special arm** – and increase its reward to $1/2 + \beta$ for some $\beta > 0$.*

For any parameter $\gamma \in (0, \frac{1}{2}]$, any algorithm that outputs a set S of $(\gamma \cdot k/12)$ arms such that with probability at least γ the special arm belongs to S requires $\frac{1}{300} \cdot \frac{\gamma^3}{\beta^2} \cdot k$ arm pulls.

Proof We give a reduction proof in the 'direct-sum' style in the same spirit of Assadi and Wang (2022). In particular, we show that if there exists an algorithm with less than $\frac{1}{300} \cdot \frac{\gamma^3}{\beta^2} \cdot k$ sample that 'traps' the special arm, we can turn it into an algorithm that contradicts the sample lower bound in Lemma 12 with $\rho = \frac{1}{2}$ and $\varepsilon = \frac{\gamma}{6}$ (which would require $\frac{1}{36} \cdot \frac{\gamma^2}{\beta^2}$ arm pulls by Lemma 12). The reduction goes as follows.

Inputs:

a) A single $\widetilde{\text{arm}}$ with the mean reward following the distribution in Lemma 12 with $\rho = \frac{1}{2}$;

b) An algorithm ALG that outputs a set S of $(\gamma \cdot k/2)$ arms such that (i). with probability at least γ, the special arm belongs to S; (ii). ALG uses less than $\frac{1}{300} \cdot \frac{\gamma^3}{\beta^2} \cdot k$ arm pulls.

Procedure:

1. With probability $(\frac{1}{2} - \frac{\gamma}{3})$, output *"reward of* $\widetilde{\text{arm}}$ *is* $\frac{1}{2} + \beta$*"*.
2. With probability $(\frac{1}{2} + \frac{\gamma}{3})$, output with the following procedure:

(i) Create $k-1$ 'dummy arms' and let their reward mean be $\frac{1}{2}$.

(ii) Put $\widetilde{\text{arm}}$ uniformly at random on index i^* among the k arms, and run ALG.

(iii) If ALG uses more than $\frac{1}{37}\cdot\frac{\gamma^2}{\beta^2}$ arm pulls on $\widetilde{\text{arm}}$, abort ALG and output *"reward of* $\widetilde{\text{arm}}$ *is* $\frac{1}{2}+\beta$*"*.

(iv) Otherwise, if the output of S contains index i^*, output *"reward of* $\widetilde{\text{arm}}$ *is* $\frac{1}{2}+\beta$*"*; if the output of S does *not* contain index i^*, output *"reward of* $\widetilde{\text{arm}}$ *is* $\frac{1}{2}$*"*.

It is straightforward to see that the algorithm never uses more than $\frac{1}{37}\cdot\frac{\gamma^2}{\beta^2}$ arm pulls on the special arm, as we directly terminate the process whenever it uses more arm pulls. It remains to verify the correctness of distinguishing the cases.

Case A): the true reward of $\widetilde{\text{arm}}$ is 1/2. Due to Line 1, there is a probability of $\frac{1}{2}-\frac{\gamma}{3}$ that the reduction never outputs "reward of $\widetilde{\text{arm}}$ is $\frac{1}{2}$". Nevertheless, we will eventually show that when the algorithm does *not* enter Line 1, the marginal correct probability is high enough to guarantee an overall $\frac{1}{2}+O(\gamma)$ correct probability.

Let s_i be the number of samples that uses on an arm with index i. Note that in this way, s_{i^*} stands for the number of samples used for $\widetilde{\text{arm}}$. Since the index of i^* is chosen uniformly at random, there is

$$\begin{aligned}
\mathbb{E}\left[s_{i^*}\right] &= \sum_{i=1}^{k}\Pr\left(i^*=i\right)\cdot\mathbb{E}\left[s_{i^*}\mid i^*=i\right] \\
&= \frac{1}{k}\cdot\sum_{i=1}^{k}\mathbb{E}\left[s_i\right] \\
&= \frac{1}{k}\cdot\mathbb{E}\left[\sum_{i=1}^{k}s_i\right] && \text{(by linearity of expectation)} \\
&\leqslant \frac{1}{300}\cdot\frac{\gamma^3}{\beta^2}. && \text{(by the sample upper bound of ALG)}
\end{aligned}$$

Therefore, by a Markov bound, we can upper-bound the probability for the special arm to use more than $\frac{1}{37}\cdot\frac{\gamma^2}{\beta^2}$ arm pulls by

$$\Pr\left(s_{i^*}\geqslant\frac{1}{37}\cdot\frac{\gamma^2}{\beta^2}\right)\leqslant\frac{\gamma}{8}.$$

As such, the probability for the reduction to false report reward as $\frac{1}{2}+\beta$ from Line (iii) is at most $\frac{\gamma}{5}$. Furthermore, in line (iv), since the arms are identical random variables and the index i^* is chosen uniformly at random, there is

$$\Pr\left(S\text{ contains index }i^*\right)=\Pr\left(S\text{ contains index }i,\,\forall i\right)=\frac{|S|}{k}=\frac{\gamma}{8}.$$

Therefore, the probability for the reduction to falsely output "reward of $\widetilde{\text{arm}}$ is $\frac{1}{2}+\beta$" through the output of ALG on line (iv) is at most $\frac{\gamma}{8}$. As such, we have

$$\Pr\left(\text{ALG outputs "reward is } \frac{1}{2}\text{"} \mid \text{reward is } \frac{1}{2},\ \ \text{Line 2 happens}\right) \geqslant 1-\frac{\gamma}{8}-\frac{\gamma}{8}=1-\frac{\gamma}{4}. \qquad \text{(by union bound)}$$

As such, the probability for the reduction to succeed when the special arm is with reward $\frac{1}{2}$ is at least:

$$\begin{aligned}
&\Pr\left(\text{ALG outputs "reward is } \frac{1}{2}\text{"} \mid \text{reward is } \frac{1}{2}\right)\\
&= \Pr\left(\text{ALG outputs "reward is } \frac{1}{2}\text{"} \mid \text{reward is } \frac{1}{2},\ \ \text{Line 2 happens}\right)\cdot \Pr\left(\text{Line 2 happens}\right)\\
&\geqslant (\frac{1}{2}+\frac{\gamma}{3})\cdot(1-\frac{\gamma}{8}-\frac{\gamma}{8})\\
&\geqslant \frac{1}{2}+\frac{\gamma}{6}. && \text{(using } \gamma \leqslant 1/2)
\end{aligned}$$

Case B): the true reward of $\widetilde{\text{arm}}$ is $1/2+\beta$. In this case, there is a probability of $(\frac{1}{2}-\frac{\gamma}{3})$ that the algorithm simply outputs "reward of $\widetilde{\text{arm}}$ is $\frac{1}{2}+\beta$" from Line 1. Furthermore, in the case of Line 2, the reduction succeed with a probability that is at least as large as γ by the guarantee of the trapping algorithm ALG. As such, the success probability in this case is at least

$$\begin{aligned}
\Pr\left(\text{ALG outputs "reward is } \frac{1}{2}+\beta\text{"} \mid \text{reward is } \frac{1}{2}+\beta\right) &\geqslant \frac{1}{2}-\frac{\gamma}{3}+(\frac{1}{2}-\frac{\gamma}{3})\cdot\gamma\\
&\geqslant \frac{1}{2}+\frac{\gamma}{6}.
\end{aligned}$$

Summarizing the cases of A) and B) establishes the correctness of the reduction for $\rho=\frac{1}{2}$ and $\varepsilon=\frac{\gamma}{6}$. By Lemma 12, the sample complexity has to be at least $\frac{1}{36}\cdot\frac{\gamma^2}{\beta^2}$, which contradicts the $\frac{1}{37}\cdot\frac{\gamma^2}{\beta^2}$ sample complexity and proves the lemma. ■

The second lemma uses a distribution similar to Lemma 14, albiet the special arm is now allowed to be "flat" – with mean reward $\frac{1}{2}$ – with probability $1-\alpha$. The lemma says that if the number of used arm pulls is small, the internal distribution (the "knowledge") of the algorithm on whether the special arm is "flat" remains close to the original, i.e., with probability $\sim(1-\alpha)$.

Lemma 15 (A sample-knowledge trade-off lemma) *Consider the following distribution $\mathcal{D}$ on $k\geqslant 1$ arms for some parameters $\alpha,\beta>0$ and $\alpha<\frac{1}{2}$:*

- ***No** case: with probability α, all except for one uniformly at random chosen arm have mean reward $1/2$, while the chosen arm have reward $1/2+\beta$;*
- ***Yes** case: with probability $1-\alpha$, all the arms have mean reward $1/2$.*

Suppose we have an algorithm that given an instance I sampled from this distribution $\mathcal{U}$ makes at most $\frac{1}{100} \cdot \frac{\gamma^2 \cdot k}{\alpha \cdot \beta^2}$ arm pulls for some $\gamma \in (0, \frac{1}{5}]$. Let Π and Π be the random variable and the realization of the transcript. Then, with probability at least $(1 - 2\gamma^{1/2})$ over the randomness of Π,

$$\Pr_I(I \text{ is a } \textbf{\textit{No}} \text{ case} \mid \Pi = \Pi) \in [\alpha - 2 \cdot \gamma^{1/2}, \alpha + 2 \cdot \gamma^{1/2}];$$
$$\Pr_I(I \text{ is a } \textbf{\textit{Yes}} \text{ case} \mid \Pi = \Pi) \in [1 - \alpha - 2 \cdot \gamma^{1/2}, 1 - \alpha + 2 \cdot \gamma^{1/2}].$$

Proof We again prove the lemma by a 'direct-sum' type of reduction. In particular, we show that for a family of arms distributed as prescribed by Lemma 15, any algorithm that learns the distribution of I with ε advantage over random guessing and s arm pulls can learn the distribution in Lemma 13 with $O(\varepsilon)$ advantage and $O(\frac{s}{k} \cdot \text{poly}(\frac{1}{\varepsilon}))$ arm pulls. This allows us to eventually build a contradiction towards Lemma 13 with $\rho = \alpha$ and $\varepsilon = 2\gamma^{1/2}$.

Inputs:

a) A single arm with mean reward following the distribution in Lemma 13 with $\rho = \alpha$;

b) An algorithm ALG that outputs $\Pr_I(I \text{ is a } \textbf{No} \text{ case} \mid \Pi = \Pi)$ as in Lemma 15.

Procedure:

1. Create $k - 1$ 'dummy arms' and let their mean reward be $\frac{1}{2}$.
2. Put the special arm uniformly at random at index i^* among the k arms, and run ALG.
3. If the special arm uses more than $\frac{1}{5} \cdot \frac{\gamma^{3/2}}{\beta^2 \alpha}$ arm pulls, stop the algorithm and output **No**.
4. Otherwise, set the probability of $\Pr(\Theta = 1 \mid \Pi = \Pi)$ in Lemma 13 (i.e., the arm is from $\text{Bern}(1/2 + \beta)$) as the same with $\Pr_I(I \text{ is a } \textbf{No} \text{ case} \mid \Pi = \Pi)$.

We focus on the case of the upper bound of $\Pr_I(I \text{ is a } \textbf{No} \text{ case} \mid \Pi = \Pi)$ since the lower bound follows from the same logic. Suppose for the purpose of contradiction that ALG uses at most $\frac{1}{100} \cdot \frac{\gamma^2 \cdot k}{\alpha \cdot \beta^2}$ arm pulls and achieves

$$\Pr_I(I \text{ is a } \textbf{No} \text{ case} \mid \Pi = \Pi) > \alpha + 2 \cdot \gamma^{1/2}.$$

By letting $\varepsilon = 2\gamma^{1/2}$, the reduction deterministically uses at most $\frac{1}{5} \cdot \frac{\gamma^{3/2}}{\beta^2 \alpha} = \frac{1}{40} \cdot \frac{\varepsilon^3}{\beta^2 \alpha} < \frac{1}{12} \cdot \frac{\varepsilon^3}{\beta^2 \alpha}$ arm pulls as we terminate whenever it uses more. We now show that with the reduction, there is

$$\Pr_\Pi\left(\Pr(\Theta = 1 \mid \Pi = \Pi) > \alpha + 2 \cdot \gamma^{1/2}\right) \geqslant 1 - 2\gamma^{1/2},$$

which leads to a contradiction with Lemma 13 with our choice of ε.

Note that I is a **No** case if and only if $\Theta = 1$ (the special arm is with mean reward $\frac{1}{2}+\beta$). As such, if the reduction reaches Line 4, then by the guarantee of the algorithm ALG, there is

$$\Pr_I(I \text{ is a } \mathbf{No} \text{ case} \mid \Pi = \Pi) = \Pr(\Theta = 1 \mid \Pi = \Pi) > \alpha + 2 \cdot \gamma^{1/2}$$

by the assumption of ALG. On the other hand, if the reduction stops by using $\frac{1}{5} \cdot \frac{\gamma^{3/2}}{\beta^2\alpha}$ arm pulls on the special arm, we show that the correct probability for the output of the **No** case is high. Note that if the instance is in the **Yes** case, the special arm is with mean reward $1/2$. Therefore, the arms for ALG are identical and independent random variables. Since the index of i^* is chosen uniformly at random, by the same arguement we used in Lemma 14, the expected number of arm pulls on the special arm is

$$\mathbb{E}\left[s_{i^*} \mid \mathbf{Yes} \text{ case}\right] \leqslant \frac{1}{100} \cdot \frac{\gamma^2}{\alpha \cdot \beta^2}.$$

As such, by a simple Markov bound, we have

$$\Pr\left(s_{i^*} \geqslant \frac{1}{5} \cdot \frac{\gamma^{3/2}}{\beta^2\alpha} \mid \mathbf{Yes} \text{ case}\right) \leqslant \frac{\gamma^{1/2}}{20}.$$

Therefore, the probability for Line 3 to output correctly output **No** case is at least

$$1 - \frac{\gamma^{1/2}}{20} > 1 - 2\,\gamma^{1/2} > 2\,\gamma^{1/2},$$

where the last inequality is by the range of γ. As such, the above implies

$$\Pr_{\Pi}\left(\Pr(\Theta = 1 \mid \Pi = \Pi) = 1\right) > 2\,\gamma^{1/2},$$

and it forms the desired contradiction. ■

Appendix D. The Full Analysis of the Multi-Pass Lower Bound

We now proceed to the main part of the proof of Theorem 1. To continue, we introduce some additional notation used in the analysis in a self-contained manner.

Additional Notation. Let $P \geqslant 2$, $C \geqslant 1$ be postitive integers, $\mathcal{D}(P, C)$ be the distribution of hard instances defined in Section 3, and ALG be a P-pass (deterministic) streaming algorithm defined as in Section 2.1. For each batch B_q, we use $I(B_q)$ to denote the *indices* in the *order of the stream* of batch q, i.e. $I(B_q) = ((P-q+1)\cdot\frac{n}{P+1}, (P-q+2)\cdot\frac{n}{P+1}]$. The batch B_q hence contains the arms $\sigma(i)$ for $i \in I(B_q)$. For any integer $p \in [P+1]$, we introduce new notation to handle variables as functions of p as follows.

- **Transcripts.** Denote Π^p and Π^p as the random variable and the realization of the transcript induced by the arm pulls *within* the p-th pass. We further define $\Pi^{1:p} := (\Pi^1, \ldots, \Pi^p)$ and $\Pi^{1:p} := (\Pi^1, \ldots, \Pi^p)$ as the random variable and the realization of the transcript among *all* of the first p-passes. For transcripts Π^p, $\Pi^{1:p}$, etc., we define batch-specific transcripts as follows. We define $\Pi^p_{\cap B_q}$ (resp. $\Pi^p_{\cap B_q}$) be the transcript induced by the arm pulls *on the arms in the q-th batch*, i.e. the result of $\text{arm}_{\sigma(i)}$ is recorded in $\Pi^p_{\cap B_q}$ if $i \in I(B_q)$. The notation generalizes to $\Pi^{1:p}_{\cap B_q}$ as well.

- **Memory.** We use M^p and M^p to denote the random variable and the realization of the memory state by the *end* of the p-th pass.
- **Sample complexity.** We similary define $\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{q+1}}$ as the *total* number of arm pulls used on the *arms* from the $(q+1)$-th batch to the $(P+1)$-th batch, i.e. when calling the sampler $\mathcal{O}$, the index $i \in \cup_{r=q+1}^{P+1} I(B_r)$ (and it is independent of j_{arrive}). Similarly, we use $\mathsf{Smp}(\text{ALG})_{B_q}$ to denote the *total* number of arm pulls used on the arms in batch q. To avoid confusion, when we talk about the total number of arm pulls in pass p, it means the cumulative number of arm pulls in the *first p passes*.

Notice that the final output of the algorithm is a deterministic function of $(\Pi^{1:P}, M^P)$.

Since we work with the expectation over the randomness of the memory and transcript, to avoid very long lines, we sometimes use

$$\mathop{\mathbb{E}}_{\Pi^{1:p-1},\mathsf{M}^{p-1}} \left[\mathsf{Smp}(\text{ALG}) \mid \Pi^{1:p-1}, M^{p-1}, \Theta_{<p} = 0\right]$$

for the full expression of $\mathbb{E}_{\mathsf{\Pi}^{1:p-1},\mathsf{M}^{p-1}} \left[\mathsf{Smp}(\text{ALG}) \mid \mathsf{\Pi}^{1:p-1} = \Pi^{1:p-1}, \mathsf{M}^{p-1} = M^{p-1}, \Theta_{<p} = 0\right]$. Other random variables that appear on the conditions follow the same rule for simplifications.

Memory- and Batch-obliviousness. We now introduce the notion of memory- and batch-obliviousness, which we will use crucially to describe the "limits of learning" for any streaming algorithms.

We say that the algorithm is **memory-oblivious** at the end of pass p if M^p contains no arm with reward strictly more than $1/2$ during *any* of the first p passes. Notice that in particular if the algorithm is memory-oblivious at the end of the P-th pass, then it cannot output the best arm in the stream. We use $\mathcal{E}^p_{\text{mem-obl}}$ to denote the *event* that ALG is memory-oblivious by the *end* of pass p. Note that the memory-oblivious event is *downward implications*: if the algorithm is memory oblivious at the end of pass p, it has to be memory-oblivious for all passes $p' < p$.

We further say that the algorithm is **batch-oblivious** at the end of pass p if given $(\Pi^{1:p}, M^p)$, and conditioning on the event of $\Theta_{<p} = 0$, for any $q > p$, the algorithm "does not know" the value of Θ_q; formally,

$$\forall\ p < q \leqslant P: \quad \Pr\left(\Theta_q = 1 \mid \mathsf{\Pi}^{1:p} = \Pi^{1:p}, \mathsf{M}^p = M^p, \Theta_{<p+1} = 0\right) \in [\frac{1}{2P} - \frac{1}{4P^2}\,,\, \frac{1}{2P} + \frac{1}{4P^2}]. \tag{4}$$

We use $\mathcal{E}^p_{\text{batch-obl}}$ to denote the *event* that ALG is batch-oblivous by the *end* of pass p.

The strategy of the proof. We will inductively show that the algorithm is going to be memory-oblivious and batch-oblivious with a large probability throughout each passes. To do so, we consider two types of possible behavior for the algorithm ALG in each pass p:

- **Conservative** case: the first case is when the algorithm decides to be "conservative" with its arm pulls in the first p passes. We show that if the probability for the algorithm to be in conservative case after the first $p-1$ passes is large, and the algorithm decides to be conservative on the first p passes, then the algorithm is going to remain memory-oblivious and batch-oblivious for the subsequent pass as well with a sufficiently large probability.

- **Radical** case: the complementary case is when the algorithm decides to make "many" arm pulls in the first p passes. In this case, we use the memory- and batch-obliviousness properties of the algorithm to show that such an algorithm is necessarily going to break the guarantee on the number of arm pulls imposed on it by Theorem 1 in some cases.

Formalizing this strategy is challenging due to the nature of guarantee of Theorem 1 on the event $\mathcal{E}_{\text{First}}(p)$ for some unknown p (rather informally speaking, since η_p is unknown to the algorithm, but also follows a certain distribution in the input). This requires a careful conditioning on various events happening in the algorithm and keeping track of the information revealed by these events.

D.1. The Conservative Case

The following lemma allows us to handle the conservative case (which is a restatement of the same lemma in Section 4).

Lemma 4 (Conservative case) *For any integer $p \in [P]$, let* ALG *be a streaming algorithm with a memory of at most $n/(20000P^3)$ arms, and assume that at the end of the pass $p-1$, the following conditions hold*

(I). The probability for $\mathcal{E}^{p-1}_{\textit{batch-obl}}, \mathcal{E}^{p-1}_{\textit{mem-obl}}$ and $\Theta_{<p} = 0$ to happen is large, i.e.,

$$\Pr\left(\mathcal{E}^{p-1}_{\textit{batch-obl}}, \mathcal{E}^{p-1}_{\textit{mem-obl}}, \Theta_{<p} = 0\right) \geqslant \left(1 - \frac{1}{2P}\right)^{10(p-1)};$$

(II). Conditioning on $\Theta_{<p} = 0$ and $\mathcal{E}^{p-1}_{\textit{batch-obl}}, \mathcal{E}^{p-1}_{\textit{mem-obl}}$, the expected number of arm pulls (over the randomness of the first p passes) is small, i.e.,

$$\mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \mathcal{E}^{p-1}_{\textit{batch-obl}}, \mathcal{E}^{p-1}_{\textit{mem-obl}}, \Theta_{<p} = 0\right] \leqslant \frac{1}{10^9} \cdot \frac{n}{\gamma_{p+1} \cdot P^{30}}.$$

Then, with probability at least $(1-1/2P)^{10p}$, at the end of pass p, we have $\Theta_{<p+1} = 0$ and the algorithm is memory- and batch-oblivious, i.e.,

$$\Pr\left(\mathcal{E}^{p}_{\textit{batch-obl}}, \mathcal{E}^{p}_{\textit{mem-obl}}, \Theta_{<p+1} = 0\right) \geqslant \left(1 - \frac{1}{2P}\right)^{10p}.$$

We prove Lemma 4 in the rest of Section D.1. To this end, we show two main technical lemmas towards the proof of the memory and batch obliviousness.

The first technical lemma characterizes a "no storing" constraint of a p-pass algorithm that satisfies the conditions as prescribed in Lemma 4. This is a "streaming version" of the offline "no trapping" result of Lemma 14.

Lemma 16 *Let $p \in [P+1]$ be a parameter, and let* ALG *be a p-pass streaming algorithm with a memory of $n/(20000P^3)$ arms. Let $q \in (p, P+1]$, and suppose the underlying instance from $\mathcal{D}(P, C)$ satisfies that batches $B_{\leqslant p}$ contain only arms with mean rewards $\frac{1}{2}$, i.e. $\Theta_{\leqslant p} = 0$.*

Furthermore, suppose the assumptions of Item (I). and Item (II). in Lemma 4 hold. Then, conditioning on $\mathcal{E}^{p-1}_{\textit{batch-obl}}, \mathcal{E}^{p-1}_{\textit{mem-obl}}, \Theta_{\leqslant p} = 0$, *the probability for* ALG *to store an arm with*

mean reward strictly more than $\frac{1}{2}$ *from* B_q *is at most* $\frac{1}{2P^2}$*, i.e., let* $\mathcal{E}^p_{mem>1/2}(q)$ *be the event that* ALG *stores the special arm of batch* q*, there is*

$$\Pr\left(\mathcal{E}^p_{mem>1/2}(q) \mid \mathcal{E}^{p-1}_{batch\text{-}obl}, \mathcal{E}^{p-1}_{mem\text{-}obl}, \Theta_{\leqslant p} = 0\right)$$
$$= \mathop{\mathbb{E}}_{\Pi^{1:p}, \mathsf{M}^p}\left[\Pr\left(\mathcal{E}^p_{mem>1/2}(q) \mid \mathcal{E}^{p-1}_{batch\text{-}obl}, \mathcal{E}^{p-1}_{mem\text{-}obl}, \Theta_{\leqslant p} = 0, \Pi^{1:p} = \Pi^{1:p}, \mathsf{M}^p = M^p\right)\right] \leqslant \frac{1}{2P^2}.$$

We explicitly write the expectation over $\Pi^{1:p}, \mathsf{M}^p$ *to emphasize the randomness over the transcript and the memory of the first* p *passes.*

Proof We first note that the randomness in the statement of Lemma 16 includes the randomness of the transcript and the memory of the first p passes and the underlying instance. We use a reduction argument from the problem in Lemma 14 to establish the desired lower bound. In particular, we show that conditioning on all the conditions in Lemma 16, if ALG can store the special arm in batch B_q with probability more than $1/2P^2$, then we can "trap" the best arm in batch q by running ALG with $\gamma = \frac{1}{100P^2}$, $k = \frac{n}{P+1}$, and $\beta = \eta_q$. The success probability is non-negligible, albeit low, and such an algorithm would require a high sample complexity. However, since we assume low sample complexity (condition Item (II).), it will lead to a contradiction with Lemma 14.

We now formalize the above intuition. We first give a detailed simulation procedure as follows.

An algorithm (reduction) for the problem in Lemma 14

Input: B_q: k arms with one *special arm* as in Lemma 14 with $\beta = \eta_q$;
Input: ALG: a streaming algorithm that stores the special arm of B_q with probability more than $1/2P^2$ conditioning on the event of Lemma 16.
Parameters: $\gamma = \frac{1}{100P^2}$ $\quad k = \frac{n}{P+1}$ $\quad \beta = \eta_q$.

1. Sample an underlying instance from $\mathcal{D}(P, C)$ for ALG as follows
   (a) Parameters η_r in $\mathcal{D}(P, C)$ as follows: let $(P - q + 1)$ parameters η_r follow the arriving order before B_q, and let $(q - 1)$ parameters η_r follow the arriving order after B_q.
   (b) Ensure the condition of $\Theta_{\leqslant p} = 0$, and sample each $\Theta_r = 1$ for $r \notin [p] \cup \{q\}$ with probability $1/2P$ (exactly as in $\mathcal{D}(P, C)$).
   (c) Sample P batches of $\frac{n}{P+1}$ arms with the above setting, and concatenate them with B_q to get the stream.
2. Run the streaming algorithm ALG on the instance:
   (a) For each pass, sample exactly as ALG does and maintain the local memory exactly as the memory of ALG.
   (b) At any point, if the number of samples is more than $\frac{1}{25000} \cdot \frac{n}{\gamma_{p+1} \cdot P^{29}}$ on batch q, abort the algorithm and output "failure".

(c) If the algorithm does *not* output "failure", at the end of the p-th pass, output all the (indices of) arms that are in B_q.

As we have discussed in Section 2.1, the offline algorithm can ignore the indices of the pass and the arriving arm in Π (by writing $*$ therein). As such, the reduction gives a valid algorithm for the problem in Lemma 14. We now lower bound the probability of $(\mathcal{E}^p_{\text{mem}>1/2}(q) \mid \Theta_{\leqslant p} = 0)$ using the probability of $\mathcal{E}^p_{\text{mem}>1/2}(q)$ *conditioning on* Item (I). of Lemma 4. Formally, we have

$$\begin{aligned}
&\Pr\left(\mathcal{E}^p_{\text{mem}>1/2}(q) \mid \Theta_{\leqslant p} = 0\right) \\
&\qquad \text{(written in the conditional form to begin with by the conditions in Lemma 14)} \\
&\geqslant \Pr\left(\mathcal{E}^p_{\text{mem}>1/2}(q) \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{\leqslant p} = 0\right) \cdot \Pr\left(\mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}} \mid \Theta_{\leqslant p} = 0\right).
\end{aligned}$$

We lower bound the second term by using the condition in Item (I). of Lemma 4 as follows:

$$\begin{aligned}
\Pr\left(\mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}} \mid \Theta_{\leqslant p} = 0\right) &= \frac{\Pr\left(\mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{\leqslant p} = 0\right)}{\Pr\left(\Theta_{\leqslant p} = 0\right)} \\
&\geqslant \Pr\left(\mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{\leqslant p} = 0\right) \\
&= \Pr\left(\mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p} = 0, \Theta_p = 0\right) \\
&= \Pr\left(\Theta_p = 0 \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right) \cdot \Pr\left(\mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p} = 0\right) \\
&\geqslant \Pr\left(\Theta_p = 0 \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right) \cdot \left(1 - \frac{1}{2P}\right)^{10(P-1)} \\
&\qquad \text{(by the condition of Item (I).)} \\
&\geqslant \left(1 - \frac{1}{4P}\right) \cdot \left(1 - \frac{1}{2P}\right)^{10(P-1)} \\
&\qquad \text{(by using the batch obliviousness)} \\
&\geqslant \frac{1}{30}. \qquad \text{(using } P \geqslant 2\text{)}
\end{aligned}$$

On the other hand, recall that by condition Item (II). of Lemma 4, there is

$$\mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p} = 0\right] \leqslant \frac{1}{10^9} \cdot \frac{n}{\gamma_{p+1} \cdot P^{30}}.$$

We bound the expected sample of $(\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{\leqslant p} = 0)$ (note the extra condition of $\Theta_p = 0$) with the batch oblivious condition:

$$\begin{aligned}
&\mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{\leqslant p} = 0\right] \\
&= \mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_p = 0, \Theta_{<p} = 0\right] \\
&\leqslant \frac{1}{\Pr\left(\Theta_p = 0 \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p} = 0\right)} \cdot \mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p} = 0\right] \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(by total expectation)} \\
&\leqslant \frac{1}{1 - 3/4P} \cdot \mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p} = 0\right] \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(by batch obliviousness)} \\
&\leqslant 2 \cdot \mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p} = 0\right] \\
&\leqslant \frac{2}{10^9} \cdot \frac{n}{\gamma_{p+1} \cdot P^{30}}.
\end{aligned}$$

Note additionally that the total arm pulls we used on batch q is a subset of the arm pulls measured by $\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}}$. Therefore, conditioning on $(\mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_q = 1, \Theta_{\leqslant p} = 0)$, with probability at least $(1 - \frac{1}{100P})$, the sample complexity does not break the limit and return "failure" (Markov bound).

We further note that our input instance has $\Theta_q = 1$ deterministically. As such, we can further lower bound the probability our reduction to store the best arm by

$$\begin{aligned}
&\Pr\left(\text{ALG stores the special arm} \mid \Theta_q = 1, \Theta_{\leqslant p} = 0\right) \\
&= \frac{\Pr\left(\text{ALG stores the special arm} \mid \Theta_{\leqslant p} = 0\right)}{\Pr\left(\Theta_q = 1 \mid \Theta_{\leqslant p} = 0\right)} \\
&\qquad\qquad\qquad\qquad\qquad (\Pr(\text{ALG stores the special arm} \mid \Theta_q = 0, \Theta_{\leqslant p} = 0) = 0) \\
&\geqslant \Pr\left(\text{ALG stores the special arm} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{\leqslant p} = 0\right) \cdot \Pr\left(\mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}} \mid \Theta_{\leqslant p} = 0\right) \\
&\geqslant \Pr\left(\mathcal{E}^{p}_{\text{mem}>1/2}(q) \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{\leqslant p} = 0\right) \cdot \Pr\left(\text{not return "failure"} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{\leqslant p} = 0\right) \\
&\qquad \cdot \Pr\left(\mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}} \mid \Theta_{\leqslant p} = 0\right) \\
&\geqslant \Pr\left(\mathcal{E}^{p}_{\text{mem}>1/2}(q) \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{\leqslant p} = 0\right) \cdot \left(1 - \frac{1}{100P}\right) \cdot \frac{1}{30}.
\end{aligned}$$

Hence, the condition of

$$\Pr\left(\mathcal{E}^{p}_{\text{mem}>1/2}(q) \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{\leqslant p} = 0\right) > \frac{1}{2P^2}$$

implies

$$\Pr\left(\text{ALG stores the special arm} \mid \Theta_{\leqslant p} = 0\right) \geqslant \left(1 - \frac{1}{100P}\right) \cdot \frac{1}{30} \cdot \frac{1}{2P^2} \geqslant \frac{1}{100P^2}.$$

The final output of the reduction is a subset of the memory of the streaming algorithm ALG, which is at most $\frac{n}{20000P^3}$. Note that for any $P \geqslant 2$, there is $300\cdot(100P^2)^3\cdot(P+1) < 25000\cdot P^{29}$. Therefore, we obtain an offline algorithm that uses at most $\frac{1}{25000}\cdot\frac{n}{\eta_q^2}\cdot\frac{1}{P^{29}} < \frac{1}{300}\cdot\frac{\gamma^3}{\beta^2}\cdot k$ arm pulls and outputs at most $\frac{n}{20000P^3} < \frac{1}{12}\cdot\frac{1}{500P^2}\cdot k$ arms (using $P \geqslant 2$) that contains the special arm with probability at least $\gamma = \frac{1}{100P^2}$. This reaches a contradiction with Lemma 14, which proves Lemma 16. ■

We now show another technical lemma that deals with the "learning" aspect of a p-pass streaming algorithm that satisfies the conditions in Lemma 4. This is similarly a streaming analogy of the offline "no learning" result of Lemma 15.

Lemma 17 *Let $p \in [P+1]$ be a parameter, and let* ALG *be a p-pass streaming algorithm with a memory of $n/(20000P^3)$ arms. Suppose the underlying instance from $\mathcal{D}(P,C)$ satisfies that batches $B_{\leqslant p}$ contain only arms with mean rewards $\frac{1}{2}$, i.e. $\Theta_{\leqslant p} = 0$. Additionally, suppose the conditions of Item (I). and Item (II). in Lemma 4 hold, and there is*

$$\Pr\left(\mathcal{E}^p_{\textit{mem-obl}}, \mathcal{E}^{p-1}_{\textit{batch-obl}}, \Theta_{\leqslant p} = 0\right) \geqslant \left(1 - \frac{1}{2P}\right)^{10(p-1)+5}.$$

Then, for any $q \in (p, P]$, with probability at least $(1-\frac{1}{2P^2})$ conditioning on $(\mathcal{E}^p_{\textit{mem-obl}}, \mathcal{E}^{p-1}_{\textit{batch-obl}}, \Theta_{\leqslant p} = 0)$, *there is*

$$\Pr\left(\Theta_q = 1 \mid \Pi^{1:p} = \Pi^{1:p}, \mathsf{M}^p = M^p, \Theta_{<p+1} = 0\right) \in [\frac{1}{2P} - \frac{1}{4P^2}, \frac{1}{2P} + \frac{1}{4P^2}].$$

Proof We only show the proof for the upper bound since the lower bound follows in the same manner. Similar to the proof of Lemma 16, we show an algorithm (reduction) from the offline Lemma 15 to the streaming algorithm as follows.

An algorithm (reduction) for the problem in Lemma 15

Input: B_q: k arms following the distribution of Lemma 15 with $\beta = \eta_q$;
Input: ALG: a streaming algorithm that with probability more than $1/2P^2$ conditioning on the conditions of Lemma 17, outputs memory and transcript $\Pi^{1:p}$ and M^p such that $\Pr(\Theta_q = 1 \mid \Pi^{1:p} = \Pi^{1:p}, \mathsf{M}^p = M^p, \Theta_{<p+1} = 0) > \frac{1}{2P} + \frac{1}{4P^2}$.
Parameters: $\gamma^{1/2} = \frac{1}{250P^2}$ $\quad k = \frac{n}{P+1}$ $\quad \beta = \eta_q$.

1. Sample an underlying instance from $\mathcal{D}(P,C)$ for ALG as follows
   (a) Parameters η_r in $\mathcal{D}(P,C)$ as follows: let $(P-q+1)$ parameters η_r follwing the arriving order before B_q, and let $(q-1)$ parameters η_r follwing the arriving order after B_q.
   (b) Ensure the condition of $\Theta_{\leqslant p} = 0$, and sample each $\Theta_r = 1$ for $r \notin [p] \cup \{q\}$ with probability $1/2P$ (exactly as in $\mathcal{D}(P,C)$ for these batches).
   (c) Sample P batches of $\frac{n}{P+1}$ arms with the above setting, and concatenate them with B_q to get the stream.

2. Run the streaming algorithm ALG on the instance:
   (a) For each pass, sample exactly as ALG does and maintain the local memory exactly as the memory of ALG.
   (b) At any point, if the number of samples is more than $\frac{1}{5} \cdot \frac{1}{10^5} \cdot \frac{n}{\gamma_{p+1} \cdot P^{30}}$ on batch q, abort the algorithm and output "failure".
   (c) If the algorithm does *not* output "failure", at the end of the p-th pass, evaluate Θ_q with maximum likelihood estimation, i.e., let the trasncript and memory of the algorithm be $(\Pi^{1:p}, M^p)$, we output the distribution of

$$\left(\Theta_q \mid \Pi^{1:p} = \Pi^{1:p}, \mathsf{M}^p = M^p, \Theta_{\leqslant p} = 0\right).$$

We show that with probability more than $2\gamma^{1/2}$, the algorithm returns $\Pr(\Theta = 1 \mid \Pi = \Pi) > \frac{1}{2P} + 2\gamma^{1/2}$. Define a pair of transcript and memory $\Pi^{1:p} = \Pi^{1:p}, \mathsf{M}^p = M^p$ as *informative* if $\Pr(\Theta_q = 1 \mid \Pi^{1:p} = \Pi^{1:p}, \mathsf{M}^p = M^p, \Theta_{<p+1} = 0) > \frac{1}{2P} + 2\gamma^{1/2}$, and define $\mathcal{E}^p_{\text{inform}}(q)$ as the *event* for the streaming algorithm to produce informative transcript and memory by the end of pass p for batch q. We first lower bound the probability of the event as follows.

$$\begin{aligned}
&\Pr\left(\mathcal{E}^p_{\text{inform}}(q) \mid \Theta_{<p+1} = 0\right) \\
&\geqslant \Pr\left(\mathcal{E}^p_{\text{inform}}(q) \mid \mathcal{E}^p_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{<p+1} = 0\right) \cdot \Pr\left(\mathcal{E}^p_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}} \mid \Theta_{<p+1} = 0\right) \\
&\geqslant \Pr\left(\mathcal{E}^p_{\text{inform}}(q) \mid \mathcal{E}^p_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{<p+1} = 0\right) \cdot \Pr\left(\mathcal{E}^p_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{<p+1} = 0\right) \\
&\geqslant \Pr\left(\mathcal{E}^p_{\text{inform}}(q) \mid \mathcal{E}^p_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{<p+1} = 0\right) \cdot \left(1 - \frac{1}{2P}\right)^{10(p-1)+5} \\
&\geqslant \Pr\left(\mathcal{E}^p_{\text{inform}}(q) \mid \mathcal{E}^p_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{<p+1} = 0\right) \cdot \frac{1}{200},
\end{aligned}$$

where the second-last line is from the condition of Lemma 17, and the last line uses $p \leqslant P$ and $P \geqslant 2$. We now provide an upper bound on the expected number of arm pulls using the condition of Item (II). in Lemma 4 with the *extra condition* of $\mathcal{E}^p_{\text{mem-obl}}$. To this end, we first upper bound $\Pr(\mathcal{E}^p_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{\leqslant p} = 0)$ with the term $\Pr(\mathcal{E}^p_{\text{mem-obl}}, \Theta_p = 0 \mid$

$\mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p} = 0)$ as follows.

$$
\begin{aligned}
&\Pr\left(\mathcal{E}^{p}_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{\leqslant p} = 0\right) \\
&= \Pr\left(\mathcal{E}^{p}_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{\leqslant p} = 0 \mid \mathcal{E}^{p-1}_{\text{mem-obl}}\right) \cdot \Pr\left(\mathcal{E}^{p-1}_{\text{mem-obl}}\right) \\
&\qquad\qquad (\mathcal{E}^{p}_{\text{mem-obl}} \text{ cannot happen if } \mathcal{E}^{p-1}_{\text{mem-obl}} \text{ does } \textit{not} \text{ happen}) \\
&\leqslant \Pr\left(\mathcal{E}^{p}_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{\leqslant p} = 0 \mid \mathcal{E}^{p-1}_{\text{mem-obl}}\right) \\
&= \Pr\left(\mathcal{E}^{p}_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{<p} = 0, \Theta_p = 0 \mid \mathcal{E}^{p-1}_{\text{mem-obl}}\right) \\
&= \Pr\left(\mathcal{E}^{p}_{\text{mem-obl}}, \Theta_p = 0 \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p} = 0\right) \cdot \Pr\left(\mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{<p} = 0 \mid \mathcal{E}^{p-1}_{\text{mem-obl}}\right) \\
&\leqslant \Pr\left(\mathcal{E}^{p}_{\text{mem-obl}}, \Theta_p = 0 \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p} = 0\right).
\end{aligned}
$$

With the above inequality, we can bound the expected samples on $\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}}$ as follows.

$$
\begin{aligned}
&\mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \mathcal{E}^{p}_{\text{mem-obl}}, \Theta_{\leqslant p} = 0\right] \\
&\leqslant \mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p} = 0\right] \cdot \frac{1}{\Pr\left(\mathcal{E}^{p}_{\text{mem-obl}}, \Theta_p = 0 \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p} = 0\right)} \\
&\leqslant \mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p} = 0\right] \cdot \frac{1}{\Pr\left(\mathcal{E}^{p}_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{\leqslant p} = 0\right)} \\
&\leqslant \mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p} = 0\right] \cdot \frac{1}{\left(1 - \frac{1}{2P}\right)^{10(p-1)+5}} \\
&\qquad\qquad \text{(by the condition in Lemma 17)} \\
&\leqslant \mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p} = 0\right] \cdot 200 \\
&\leqslant \frac{1}{5} \cdot \frac{1}{10^6} \cdot \frac{n}{\gamma_{p+1} \cdot P^{30}}. \qquad \text{(using Item (II). in Lemma 4)}
\end{aligned}
$$

Once again, note that the total arm pulls we used on batch q is a subset of the arm pulls measured by $\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}}$. Therefore, by the Markov bound, with probability at least $9/10$ *conditioning on* the events of $\mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \mathcal{E}^{p}_{\text{mem-obl}}, \Theta_{\leqslant p} = 0$, the algorithm will *not* return failure.

Note that by running the reduction, the offline algorithm has access of $\Pi^{1:p}$ and M^P by the end of pass p. Hence, if we have

$$\Pr(\Theta_q = 1 \mid \Pi^{1:p} = \Pi^{1:p}, \mathsf{M}^p = M^p, \Theta_{<p+1} = 0) > \frac{1}{2P} + 2\gamma^{1/2},$$

there is

$$\Pr(\Theta = 1 \mid \Pi = \Pi) > \frac{1}{2P} + 2\gamma^{1/2}$$

from the perspective of the offline algorithm. Therefore, we have that

$$\begin{aligned}\Pr\left(\Pr(\Theta=1\mid\Pi=\Pi)>\frac{1}{2P}+2\gamma^{1/2}\right)&=\Pr\left(\Pr(\Theta=1\mid\Pi=\Pi)>\frac{1}{2P}+2\gamma^{1/2}\mid\Theta_{<p+1}=0\right)\\&\qquad(\Theta_{<p+1}=0\text{ is ensured in the instances})\\&=\Pr\left(\mathcal{E}^p_{\text{inform}}(q)\mid\Theta_{<p+1}=0\right).\end{aligned}$$

As such, we can combine this with the lower bound of $\Pr(\mathcal{E}^p_{\text{inform}}(q)\mid\Theta_{<p+1}=0)$ to get that if we have

$$\Pr\left(\mathcal{E}^p_{\text{inform}}(q)\mid\mathcal{E}^p_{\text{mem-obl}},\mathcal{E}^{p-1}_{\text{batch-obl}},\Theta_{<p+1}=0\right)\geqslant\frac{1}{2P^2},$$

it implies that

$$\begin{aligned}\Pr\left(\Pr(\Theta=1\mid\Pi=\Pi)>\frac{1}{2P}+2\gamma^{1/2}\right)&=\Pr\left(\mathcal{E}^p_{\text{inform}}(q)\mid\Theta_{<p+1}=0\right)\\&\geqslant\Pr\left(\mathcal{E}^p_{\text{inform}}(q)\mid\mathcal{E}^p_{\text{mem-obl}},\mathcal{E}^{p-1}_{\text{batch-obl}},\Theta_{<p+1}=0\right)\cdot\frac{1}{200}\\&\geqslant\frac{1}{2P^2}\cdot\frac{1}{200}\cdot\frac{9}{10}\geqslant\frac{1}{500P^2}.\end{aligned}$$

Furthermore, the algorithm uses at most $\frac{1}{5}\cdot\frac{1}{10^5}\cdot\frac{n}{\gamma_{p+1}\cdot P^{30}}$ arm pulls on batch B_q.

Let $2\gamma^{1/2}=\frac{1}{500P^2}$, we have the guarantee of the knowledge on Θ (of the offline algorithm) becomes $\Pr(\Theta=1\mid\Pi=\Pi)>\frac{1}{2P}+\frac{1}{4P^2}>\frac{1}{2P}+\frac{1}{500P^2}$. Therefore, such a algorithm should require $\frac{1}{100}\cdot\frac{\gamma^2\cdot k}{\alpha\cdot\beta^2}$ arm pulls. For any $P\geqslant 2$, there is $\frac{100\cdot(250P^2)^4\cdot(P+1)}{2P}<5\cdot10^5\cdot P^{30}$, which implies $\frac{1}{5}\cdot\frac{1}{10^5}\cdot\frac{n}{\gamma_{p+1}\cdot P^{30}}<\frac{1}{100}\cdot\frac{\gamma^2\cdot k}{\alpha\cdot\beta^2}$ arm pulls ($\beta=\eta_q\leqslant\sqrt{\gamma_{p+1}}$). This forms a contradiction with Lemma 15, which proves the lemma. ■

We are ready to proceed to the proof of the main claims in Lemma 4 as follows.

Proof of Lemma 4 We first lower bound the probability of $\Pr(\mathcal{E}^p_{\text{batch-obl}},\mathcal{E}^p_{\text{mem-obl}},\Theta_{<p+1}=0)$ as a function of $\Pr(\mathcal{E}^p_{\text{batch-obl}},\mathcal{E}^p_{\text{mem-obl}}\mid\mathcal{E}^{p-1}_{\text{batch-obl}},\mathcal{E}^{p-1}_{\text{mem-obl}},\Theta_{<p+1}=0)$. To this end, we lower bound $\Pr(\mathcal{E}^p_{\text{batch-obl}},\mathcal{E}^p_{\text{mem-obl}},\Theta_{<p+1}=0)$ as follows:

$$\begin{aligned}&\Pr\left(\mathcal{E}^p_{\text{batch-obl}},\mathcal{E}^p_{\text{mem-obl}},\Theta_{<p+1}=0\right)\\&\geqslant\Pr\left(\mathcal{E}^p_{\text{batch-obl}},\mathcal{E}^p_{\text{mem-obl}},\Theta_{<p+1}=0\mid\mathcal{E}^{p-1}_{\text{batch-obl}},\mathcal{E}^{p-1}_{\text{mem-obl}},\Theta_{<p}=0\right)\cdot\Pr\left(\mathcal{E}^{p-1}_{\text{batch-obl}},\mathcal{E}^{p-1}_{\text{mem-obl}},\Theta_{<p}=0\right),\end{aligned}$$

in which the first term of the right hand side can be factored to

$$\begin{aligned}&\Pr\left(\mathcal{E}^p_{\text{batch-obl}},\mathcal{E}^p_{\text{mem-obl}},\Theta_{<p+1}=0\mid\mathcal{E}^{p-1}_{\text{batch-obl}},\mathcal{E}^{p-1}_{\text{mem-obl}},\Theta_{<p}=0\right)\\&=\Pr\left(\Theta_{<p+1}=0\mid\mathcal{E}^{p-1}_{\text{batch-obl}},\mathcal{E}^{p-1}_{\text{mem-obl}},\Theta_{<p}=0\right)\cdot\Pr\left(\mathcal{E}^p_{\text{batch-obl}},\mathcal{E}^p_{\text{mem-obl}}\mid\mathcal{E}^{p-1}_{\text{batch-obl}},\mathcal{E}^{p-1}_{\text{mem-obl}},\Theta_{<p+1}=0\right)\\&=\Pr\left(\Theta_p=0\mid\mathcal{E}^{p-1}_{\text{batch-obl}},\mathcal{E}^{p-1}_{\text{mem-obl}},\Theta_{<p}=0\right)\cdot\Pr\left(\mathcal{E}^p_{\text{batch-obl}},\mathcal{E}^p_{\text{mem-obl}}\mid\mathcal{E}^{p-1}_{\text{batch-obl}},\mathcal{E}^{p-1}_{\text{mem-obl}},\Theta_{<p+1}=0\right)\\&\qquad(\text{expanding the condition of }\Theta_{<p+1}=0)\\&\geqslant\left(1-\frac{1}{2P}-\frac{1}{4P^2}\right)\cdot\Pr\left(\mathcal{E}^p_{\text{batch-obl}},\mathcal{E}^p_{\text{mem-obl}}\mid\mathcal{E}^{p-1}_{\text{batch-obl}},\mathcal{E}^{p-1}_{\text{mem-obl}},\Theta_{<p+1}=0\right)\\&\qquad(\text{by batch-obliviousness by the end of pass }(p-1))\\&\geqslant(1-\frac{3}{4P})\cdot\Pr\left(\mathcal{E}^p_{\text{batch-obl}},\mathcal{E}^p_{\text{mem-obl}}\mid\mathcal{E}^{p-1}_{\text{batch-obl}},\mathcal{E}^{p-1}_{\text{mem-obl}},\Theta_{<p+1}=0\right).\end{aligned}$$

Therefore, we obtain a valid lower bound for $\Pr\left(\mathcal{E}^p_{\text{batch-obl}}, \mathcal{E}^p_{\text{mem-obl}}, \Theta_{<p+1}=0\right)$ as

$$\begin{aligned}
&\Pr\left(\mathcal{E}^p_{\text{batch-obl}}, \mathcal{E}^p_{\text{mem-obl}}, \Theta_{<p+1}=0\right)\\
&\geqslant (1-\frac{3}{4P})\cdot \Pr\left(\mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p}=0\right)\cdot \Pr\left(\mathcal{E}^p_{\text{batch-obl}}, \mathcal{E}^p_{\text{mem-obl}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p+1}=0\right)\\
&\geqslant (1-\frac{3}{4P})\cdot \left(1-\frac{1}{2P}\right)^{10(p-1)}\cdot \Pr\left(\mathcal{E}^p_{\text{batch-obl}}, \mathcal{E}^p_{\text{mem-obl}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p+1}=0\right).
\end{aligned}$$

As such, we only need to lower boud the last term, i.e., the term of $\Pr(\mathcal{E}^p_{\text{batch-obl}}, \mathcal{E}^p_{\text{mem-obl}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p+1}=0)$. To this end, we further write the probability as

$$\begin{aligned}
&\Pr\left(\mathcal{E}^p_{\text{batch-obl}}, \mathcal{E}^p_{\text{mem-obl}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p+1}=0\right)\\
&= \Pr\left(\mathcal{E}^p_{\text{batch-obl}} \mid \mathcal{E}^p_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p+1}=0\right)\cdot \Pr\left(\mathcal{E}^p_{\text{mem-obl}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p+1}=0\right).
\end{aligned}$$

We start with bounding the term $\Pr\left(\mathcal{E}^p_{\text{mem-obl}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p+1}=0\right)$, i.e., the memory-oblivious proof.

Memory oblivious proof. For each batch $q \in [P+1]$, we define the following event.

$\mathcal{E}^p_{\text{mem}>1/2}(q)$: the event that $\mathsf{Mem}(\text{ALG})$ contains an arm with mean reward more than $\frac{1}{2}$ from arms in batch B_q.

Recall that $\mathcal{E}^p_{\text{mem-obl}}$ is the *event* that ALG is memory-oblivious *after* pass p. By a simple union bound, we can bound the probability for $\mathcal{E}^p_{\text{mem-obl}}$ *not* to happen, i.e., the memory contains at least one arm with reward strictly more than $\frac{1}{2}$, as follows.

$$\begin{aligned}
&\Pr\left(\neg\,\mathcal{E}^p_{\text{mem-obl}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p+1}=0\right)\\
&\leqslant \sum_{q\in[P+1]} \Pr\left(\mathcal{E}^p_{\text{mem}>1/2}(q) \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p+1}=0\right).
\end{aligned}$$

It suffices to upper bound each conditional probability term of $\mathcal{E}^p_{\text{mem}>1/2}(q)$. To this end, we bound terms for q of different types.

The case of $q \in (p, P+1]$. In this case, we might have $\Theta_{P+1}=1$ for the batch B_q. As such, we can use Lemma 16 to obtain that

$$\begin{aligned}
&\Pr\left(\mathcal{E}^p_{\text{mem}>1/2}(P+1) \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p+1}=0\right)\\
&= \mathop{\mathbb{E}}_{\Pi^{1:p},\mathsf{M}^p}\left[\Pr\left(\mathcal{E}^p_{\text{mem}>1/2}(P+1) \mid \Pi^{1:p}=\Pi^{1:p}, \mathsf{M}^p=M^p, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p+1}=0\right)\right]\\
&\leqslant \frac{1}{2P^2}. \qquad \text{(using Lemma 16)}
\end{aligned}$$

In particular, the last line uses Lemma 16 by the conditions of a). conditions Items (I). and (II). of Lemma 4 and b). the underlying instance satisfied $\Theta_{\leqslant p}=0$. These conditions exactly satisfy the requirements of Lemma 16.

The case of $q \leqslant p$. Note that we have conditioned on the event that $\Theta_{<p+1} = 0$. Therefore, we always have

$$\Pr\left(\mathcal{E}^p_{\text{mem}>1/2}(q) \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p+1} = 0\right) = 0.$$

Summarizing the case analysis for *memory obliviousness*. By the above cases analysis, we can obtain that

$$\begin{aligned}&\Pr\left(\neg\,\mathcal{E}^p_{\text{mem-obl}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p+1} = 0\right)\\ &\leqslant \underbrace{\frac{1}{2P^2}}_{q=P+1} + \underbrace{(P-p+1)\cdot\frac{1}{2P^2}}_{q\in(p,P]} \leqslant \frac{1}{P}.\end{aligned}$$

Therefore, using the conditions Item (I). and Item (II). of Lemma 4 and conditioning on $\Theta_{<p+1} = 0$, the probability for ALG to be memory-oblivious by the end of the p-th pass is at least $(1-\frac{1}{P}) \geqslant (1-\frac{1}{2P})^3$ (holds for every $P \geqslant 2$), i.e.,

$$\Pr\left(\mathcal{E}^p_{\text{mem-obl}} \mid \mathcal{E}^{p-1}_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{\leqslant p} = 0\right) \geqslant \left(1-\frac{1}{2P}\right)^3. \tag{5}$$

Batch oblivious proof. We now proceed to the proof of the batch-oblivious property, which completes the building blocks for the proof of Lemma 4. We first note that by our analysis for the memory obliviousness, we have

$$\begin{aligned}&\Pr\left(\mathcal{E}^p_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{\leqslant p} = 0\right)\\
&\geqslant \Pr\left(\mathcal{E}^p_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{\leqslant p} = 0\right)\\
&\geqslant \Pr\left(\mathcal{E}^p_{\text{mem-obl}} \mid \mathcal{E}^{p-1}_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{\leqslant p} = 0\right)\cdot\Pr\left(\mathcal{E}^{p-1}_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{\leqslant p} = 0\right)\\
&\geqslant \Pr\left(\mathcal{E}^p_{\text{mem-obl}} \mid \mathcal{E}^{p-1}_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{\leqslant p} = 0\right)\cdot\Pr\left(\Theta_{\leqslant p} = 0 \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p} = 0\right)\\
&\qquad\qquad\cdot\Pr\left(\mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p} = 0\right)\\
&\geqslant \left(1-\frac{1}{2P}\right)^3\cdot\Pr\left(\Theta_p = 0 \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p} = 0\right)\cdot\Pr\left(\mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p} = 0\right)\\
&\qquad\qquad\text{(by Eq (5) and the definition of } \Theta_{\leqslant p})\\
&\geqslant \left(1-\frac{1}{2P}\right)^3\cdot\left(1-\frac{3}{4P}\right)\cdot\Pr\left(\mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p} = 0\right)\qquad\text{(by batch obliviousness)}\\
&\geqslant \left(1-\frac{1}{2P}\right)^3\cdot\left(1-\frac{3}{4P}\right)\cdot\left(1-\frac{1}{2P}\right)^{10(p-1)}\\
&\geqslant \left(1-\frac{1}{2P}\right)^{10(p-1)+5}.\qquad\text{(using } P \geqslant 2)\end{aligned}$$

As such, the condition for Lemma 17 is satisfied. Define $\mathcal{E}^p_{\text{inform}}(q)$ as the *event* for the streaming algorithm to produce a pair of memory and transcript $\Pi^{1:p} = \Pi^{1:p}, \mathsf{M}^p = M^p$ such

that $\Pr(\Theta_q = 1 \mid \Pi^{1:p} = \Pi^{1:p}, \mathsf{M}^p = M^p, \Theta_{<p+1} = 0) > \frac{1}{2P} + \frac{1}{4P^2}$ by the end of pass p for batch q (in the same way of Lemma 17). By applying Lemma 17, we have

$$\Pr\left(\mathcal{E}^p_{\text{inform}}(q) \mid \mathcal{E}^p_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{\leqslant p} = 0\right) \leqslant \frac{1}{2P^2}.$$

Therefore, by a union bound, we have that

$$\Pr\left(\mathcal{E}^p_{\text{batch-obl}} \mid \mathcal{E}^p_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{\leqslant p} = 0\right) \geqslant \left(1 - \frac{1}{2P}\right).$$

Finalizing the proof of Lemma 4. Recall that in the begining of the proof, we have shown that

$$\begin{aligned}
&\Pr\left(\mathcal{E}^p_{\text{batch-obl}}, \mathcal{E}^p_{\text{mem-obl}}, \Theta_{<p+1} = 0\right) \\
&\geqslant (1 - \frac{3}{4P}) \cdot \left(1 - \frac{1}{2P}\right)^{10(p-1)} \cdot \Pr\left(\mathcal{E}^p_{\text{batch-obl}}, \mathcal{E}^p_{\text{mem-obl}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p+1} = 0\right) \\
&= (1 - \frac{3}{4P}) \cdot \left(1 - \frac{1}{2P}\right)^{10(p-1)} \cdot \Pr\left(\mathcal{E}^p_{\text{batch-obl}} \mid \mathcal{E}^p_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p+1} = 0\right) \\
&\qquad \cdot \Pr\left(\mathcal{E}^p_{\text{mem-obl}} \mid \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}, \Theta_{<p+1} = 0\right).
\end{aligned}$$

Since $\mathcal{E}^p_{\text{mem-obl}}$ implies $\mathcal{E}^{p-1}_{\text{mem-obl}}$, we can bound the above chain of inequalities as

$$\begin{aligned}
&\Pr\left(\mathcal{E}^p_{\text{batch-obl}}, \mathcal{E}^p_{\text{mem-obl}}, \Theta_{<p+1} = 0\right) \\
&\geqslant (1 - \frac{3}{4P}) \cdot \left(1 - \frac{1}{2P}\right)^{10(p-1)} \cdot \left(1 - \frac{1}{2P}\right)^{3} \cdot \Pr\left(\mathcal{E}^p_{\text{batch-obl}} \mid \mathcal{E}^p_{\text{mem-obl}}, \mathcal{E}^{p-1}_{\text{batch-obl}}, \Theta_{<p+1} = 0\right) \\
&\qquad \text{(the analysis for memory obliviousness)} \\
&\geqslant (1 - \frac{3}{4P}) \cdot \left(1 - \frac{1}{2P}\right)^{10(p-1)} \cdot \left(1 - \frac{1}{2P}\right)^{3} \cdot \left(1 - \frac{1}{2P}\right) \\
&\qquad \text{(analysis for batch obliviousness)} \\
&\geqslant \left(1 - \frac{1}{2P}\right)^{10p},
\end{aligned}$$

as desired in the lemma statement. ∎

D.2. The Radical Case

We now focus on the radical case when the algorithm makes "too many" arm pulls for the "early batches" in the first p passes while being *oblivious* to these batches. We remind the readers of the main lemma with a restatement of the lemma in Section 4.

Lemma 5 (Radical case) *For any integer $p \in [P]$, suppose a streaming algorithm* ALG *is memory- and batch-oblivious at the end of the pass $p-1$, and that the underlying instance satisfies $\Theta_{<p} = 0$. Additionally, suppose*

$$\mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \mathcal{E}^{p-1}_{\textit{batch-obl}}, \mathcal{E}^{p-1}_{\textit{mem-obl}}, \Theta_{<p} = 0\right] > \frac{1}{10^9} \cdot \frac{n}{\gamma_{p+1} \cdot P^{30}};$$

then,

$$\mathbb{E}\left[\mathsf{Smp}(\text{ALG}) \mid \mathcal{E}^{p-1}_{\textit{batch-obl}}, \mathcal{E}^{p-1}_{\textit{mem-obl}}, \mathcal{E}_{\text{First}}(p)\right] > 20000 \cdot C \cdot \frac{n}{\eta_p^2}.$$

Note that the statement of Lemma 5 does *not* use the exact "symmetric" condition of Lemma 4 – we write in this way on purpose, and its usage will be clear in Section D.3.

We prove Lemma 5 for the rest of Section D.2. For technical reasons, for the proof of Lemma 5, we assume w.log. that on the p-th pass, the arm pulls of $\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}}$ are conducted *before* j_{arrive} enters batch p, i.e., before the arrival of the arms in the batch p. For any algorithm that does *not* satisfy this property, we can re-arrange the order of arm pulls on the p-th pass without changing the total number of arm pulls.

We first show a technical claim that conditioning on any (fixed) transcript and the memory by the end of the $(p-1)$-th pass, the knowledge of the algorithm on Θ_q is *independent* of the arm pulls we used in $\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{q+1}}$. Conceptually, the claim asserts the simple fact that the arm pulls induced on arms outside q, conditioning on the transcript of *all* previous passes, has nothing to do with the algorithm's knowledge for Θ_q. We also explicitly use the conditions of $\mathcal{E}^{p-1}_{\text{batch-obl}}$ and $\mathcal{E}^{p-1}_{\text{mem-obl}}$ for technical reason that will be clear later.

Claim D.1 *For any integer $p \in [P]$, let $(\Pi^{1:p-1}, M^{p-1})$ be* any *pair of transcript and memory. Then, for any realization of $\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{q+1}} = s$, i.e., the samples on batches arriving* before *q, there is*

$$\begin{aligned}
&\Pr\left(\Theta_q = 1 \mid \mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{q+1}} = s, \Pi^{1:p-1} = \Pi^{1:p-1}, \mathsf{M}^{p-1} = M^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\textit{batch-obl}}, \mathcal{E}^{p-1}_{\textit{mem-obl}}\right) \\
&= \Pr\left(\Theta_q = 1 \mid \Pi^{1:p-1} = \Pi^{1:p-1}, \mathsf{M}^{p-1} = M^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\textit{batch-obl}}, \mathcal{E}^{p-1}_{\textit{mem-obl}}\right).
\end{aligned}$$

Proof The proof is an application of the data processing inequality (Proposition 10) with a similar flavor of the rectangle property in communication protocols. Concretely, we want to prove that

$$\mathbb{I}\left(\Theta_q; \Pi^p_{\cap B_{P+1:q+1}} \mid \Pi^{1:p-1}, \mathsf{M}^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right) = 0. \tag{6}$$

Furthermore, observe that $\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{q+1}}$ is a determistic function of $\Pi^p_{\cap B_{P+1:q+1}}$ *conditioning on* $(\Pi^{1:p-1}, \mathsf{M}^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}})$. Therefore, we have

$$\begin{aligned}
&\mathbb{I}\left(\Theta_q; \mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{q+1}} \mid \Pi^{1:p-1}, \mathsf{M}^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right) \\
&\leqslant \mathbb{I}\left(\Theta_q; \Pi^p_{\cap B_{P+1:q+1}} \mid \Pi^{1:p-1}, \mathsf{M}^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right) = 0,
\end{aligned}$$

where the first inequality follows from the data-processing inequality (Proposition 10). Furthermore, since mutual information is non-negative, the above implies that for any realization $(\Pi^{1:p-1}, M^{p-1})$, there is

$$\mathbb{I}\left(\Theta_q; \mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{q+1}} \mid \Pi^{1:p-1} = \Pi^{1:p-1}, \mathsf{M}^{p-1} = M^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right) = 0.$$

Therefore, by Proposition 11, we have that

$$\Pr\left(\Theta_q = 1 \mid \mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{q+1}} = s, \Pi^{1:p-1} = \Pi^{1:p-1}, \mathsf{M}^{p-1} = M^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right)$$
$$= \Pr\left(\Theta_q = 1 \mid \Pi^{1:p-1} = \Pi^{1:p-1}, \mathsf{M}^{p-1} = M^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right),$$
(by the zero mutual information and Proposition 11)

which will reach our desired conclusion.

For the rest of this proof, we aim to establish Eq (6). To this end, we introduce the random variable for the memory inside the p-th pass, we use $\mathsf{M}^p_{>q}$ (resp. $M^p_{>q}$) to denote the random variable (resp. the realization) of the memory state when the last time j_{arrive} is smaller than all the indices in the q-th batch, i.e. the process for the memory state changing in the p-th pass can be denoted as $\mathsf{M}^p_{>P} \rightarrow \mathsf{M}^p_{>P-1} \rightarrow \ldots \rightarrow \mathsf{M}^p_{>1} \rightarrow \mathsf{M}^p_{>0} = \mathsf{M}^p$. We further define B_q as the random variable for the *arms* in batch q, and $\mathsf{B}_{>q}$ as the random variable for the *arms* in batches $(q, P+1]$.

For the ease of analysis, we use the following simple trick to *order* the arm pulls induced by $\Pi^p_{\cap B_{P+1:q+1}}$. In particular, we create a "imaginary" process that conducts all samples on batch q *before* j_{arrive} visits the batch $(q-1)$ (the next batch in the order of the stream). Note that the ordering does *not* change the value of $\mathbb{I}(\Theta_q; \Pi^p_{\cap B_{P+1:q+1}} \mid \Pi^{1:p-1}, \mathsf{M}^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}})$ since the transcript is permutation-invariate and the arm pulls on $\Pi^p_{\cap B_{P+1:q+1}}$ are conducted before j_{arrive} reaches B_q.

We start from batch $P+1$ to "inductively" prove the conditional independence between Θ_q and $\Pi^p_{\cap B_{P+1:r}}$ for $r \in (q, P+1]$. Specifically, we first use chain rule of mutual information to upper-bound the left-hand side of Eq (6) as follows.

$$\mathbb{I}\left(\Theta_q; \Pi^p_{\cap B_{P+1:q+1}} \mid \Pi^{1:p-1}, \mathsf{M}^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right)$$
$$= \mathbb{I}\left(\Theta_q; \Pi^p_{\cap B_{P+1}}, \Pi^p_{\cap B_{P:q+1}} \mid \Pi^{1:p-1}, \mathsf{M}^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right)$$
$$= \mathbb{I}\left(\Theta_q; \Pi^p_{\cap B_{P+1}} \mid \Pi^{1:p-1}, \mathsf{M}^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right)$$
$$+ \mathbb{I}\left(\Theta_q; \Pi^p_{\cap B_{P:q+1}} \mid \Pi^{1:p-1}, \mathsf{M}^{p-1}, \Pi^p_{\cap B_{P+1}}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right).$$
(by chain rule)

We now need to further "peel off" random variables from $\mathbb{I}(\Theta_q; \Pi^p_{\cap B_{P:q+1}} \mid \Pi^{1:p-1}, \mathsf{M}^{p-1}, \Pi^p_{\cap B_{P+1}}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}})$ and move the conditions "forward" in terms of the batches. To this end, consider the random variable $\mathsf{M}^p_{>P}$, and we observe that

$$\mathsf{M}^p_{>P} \perp \Theta_q \mid \Pi^{1:p-1}, \mathsf{M}^{p-1}, \Pi^p_{\cap B_{P+1}}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}$$

since the memory state is uniquely determined *after* the transcript of $\Pi^p_{\cap B_{P+1}}$ is introduced to $\Pi^{1:p-1}, \mathsf{M}^{p-1}$ (and since we use the trick to order the transcripts).

Therefore, we can further write the term $\mathbb{I}(\Theta_q; \Pi^p_{\cap B_{P:q+1}} \mid \Pi^{1:p-1}, \mathsf{M}^{p-1}, \Pi^p_{\cap B_{P+1}}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}})$ as follows.

$$
\begin{aligned}
&\mathbb{I}\left(\Theta_q; \Pi^p_{\cap B_{P:q+1}} \mid \Pi^{1:p-1}, \mathsf{M}^{p-1}, \Pi^p_{\cap B_{P+1}}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right) \\
&\leqslant \mathbb{I}\left(\Theta_q; \Pi^p_{\cap B_{P:q+1}} \mid \Pi^{1:p-1}, \Pi^p_{\cap B_{P+1}}, \mathsf{M}^{p-1}, \mathsf{M}^p_{>P}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right) \\
&\qquad\qquad \text{(condition on independent random variable does not decrease MI)} \\
&= \mathbb{I}\left(\Theta_q; \Pi^p_{\cap B_P}, \Pi^p_{\cap B_{P-1:q+1}} \mid \Pi^{1:p-1}, \Pi^p_{\cap B_{P+1}}, \mathsf{M}^{p-1}, \mathsf{M}^p_{>P}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right) \\
&= \mathbb{I}\left(\Theta_q; \Pi^p_{\cap B_P}, \Pi^p_{\cap B_{P-1:q+1}} \mid \Pi^{1:p-1}, \Pi^p_{\cap B_{P+1}}, \mathsf{M}^p_{>P}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right) \\
&\leqslant \mathbb{I}\left(\Theta_q; \Pi^p_{\cap B_P} \mid \Pi^{1:p-1}, \mathsf{M}^p_{>P}, \Pi^p_{\cap B_{P+1}}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right) \\
&\qquad + \mathbb{I}\left(\Theta_q; \Pi^p_{\cap B_{P-1:q+1}} \mid \Pi^{1:p-1}, \mathsf{M}^p_{>P}, \Pi^p_{\cap B_{P+1:P}}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right). \\
&\qquad\qquad \text{(by chain rule)}
\end{aligned}
$$

Therefore, we can keep performing the above steps, and obtain that:

$$
\begin{aligned}
&\mathbb{I}\left(\Theta_q; \Pi^p_{\cap B_{P+1:q+1}} \mid \Pi^{1:p-1}, \mathsf{M}^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right) \\
&= \sum_{r=P+1}^{q+1} \mathbb{I}\left(\Theta_q; \Pi^p_{\cap B_r} \mid \Pi^{1:p-1}, \mathsf{M}^p_{>r}, \Pi^p_{\cap B_{P+1:r+1}}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right) \\
&\leqslant \sum_{r=P+1}^{q+1} \mathbb{I}\left(B_q; \Pi^p_{\cap B_r} \mid \Pi^{1:p-1}, \mathsf{M}^p_{>r}, \Pi^p_{\cap B_{P+1:r+1}}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right),
\end{aligned}
$$

where the last inequality comes from the fact that Θ_q is a deterministic function of B_q and by using Proposition 10.

Observe that at each step, in the process with our ordering, the random variable $\Pi^p_{\cap B_r}$ is *determined* by the conditions $\Pi^{1:p-1}, \mathsf{M}^p_{>r}, \Pi^p_{\cap B_{P+1:r+1}}$. As such, we have that

$$
B_q \perp \Pi^p_{\cap B_r} \mid \Pi^{1:p-1}, \mathsf{M}^p_{>r}, \Pi^p_{\cap B_{P+1:r+1}}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}},
$$

which implies

$$
\begin{aligned}
&\mathbb{I}\left(\Theta_q; \Pi^p_{\cap B_{P+1:q+1}} \mid \Pi^{1:p-1}, \mathsf{M}^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right) \\
&\leqslant \sum_{r=P+1}^{q+1} \mathbb{I}\left(B_q; \Pi^p_{\cap B_r} \mid \Pi^{1:p-1}, \mathsf{M}^p_{>r}, \Pi^p_{\cap B_{P+1:r+1}}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right) = 0,
\end{aligned}
$$

which is as desired by Eq (6). ∎

We now proceed to the main technical lemma to prove Lemma 5: we show that conditioning on any transcript and memory that satisfies the assumptions of Lemma 5, the expected number of samples for $\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}}$ has to be high even if we add the *extra condition* of $\Theta_p = 1$. Note that a standard total expectation calculation only leads to the reverse direction, and the correctness of our case crucially relies on the lower bound for batch-obliviousness.

Lemma 18 *Let $(\Pi^{1:p-1}, M^{p-1})$ be a pair of transcript and memory of a streaming algorithm after $(p-1)$ passes, there is*

$$\mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \Pi^{1:p-1} = \Pi^{1:p-1}, \mathsf{M}^{p-1} = M^{p-1}, \Theta_{<p} = 0, \Theta_p = 1, \mathcal{E}^{p-1}_{\textit{batch-obl}}, \mathcal{E}^{p-1}_{\textit{mem-obl}}\right]$$
$$\geqslant \frac{1}{2} \cdot \mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \Pi^{1:p-1} = \Pi^{1:p-1}, \mathsf{M}^{p-1} = M^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\textit{batch-obl}}, \mathcal{E}^{p-1}_{\textit{mem-obl}}\right].$$

Proof To avoid clutter, for the given pair of transcript and memory $(\Pi^{1:p-1}, M^{p-1})$ that satisfies the lemma statement, we define random variable

$$\mathcal{S} := \mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \Pi^{1:p-1} = \Pi^{1:p-1}, \mathsf{M}^{p-1} = M^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}$$

as a notation that is self-contain in this proof. In this way, by picking realizations for $\mathcal{S} = s$, we mean $\left(\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} = s \mid \Pi^{1:p-1} = \Pi^{1:p-1}, \mathsf{M}^{p-1} = M^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right)$. By Bayes' rule, for any realization of $\mathcal{S} = s$, we have

$$\Pr\left(\mathcal{S} = s \mid \Theta_p = 1\right) = \frac{\Pr\left(\Theta_p = 1 \mid \mathcal{S} = s\right) \cdot \Pr\left(\mathcal{S} = s\right)}{\Pr\left(\Theta_p = 1\right)}.$$

As such, by using Claim D.1 with $q = p$, we have that

$$\begin{aligned}
&\Pr\left(\Theta_p = 1 \mid \mathcal{S} = s\right) \\
&= \Pr\left(\Theta_p = 1 \mid \mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} = s, \Pi^{1:p-1} = \Pi^{1:p-1}, \mathsf{M}^{p-1} = M^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right) \\
&= \Pr\left(\Theta_p = 1 \mid \Pi^{1:p-1} = \Pi^{1:p-1}, \mathsf{M}^{p-1} = M^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right).
\end{aligned}$$

Therefore, for any choice of s, we have the bound

$$\begin{aligned}
&\Pr\left(\mathcal{S} = s \mid \Theta_p = 1\right) \\
&= \frac{\Pr\left(\Theta_p = 1 \mid \Pi^{1:p-1} = \Pi^{1:p-1}, \mathsf{M}^{p-1} = M^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right) \cdot \Pr\left(\mathcal{S} = s\right)}{\Pr\left(\Theta_p = 1\right)} \\
&\geqslant \frac{\left(\frac{1}{2P} - \frac{1}{4P^2}\right)}{\Pr\left(\Theta_q = 1\right)} \cdot \Pr\left(\mathcal{S} = s\right) && \text{(by the assumption of batch-obliviousness)} \\
&\geqslant \frac{1}{2} \cdot \Pr\left(\mathcal{S} = s\right).
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
&\mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \Pi^{1:p-1} = \Pi^{1:p-1}, \mathsf{M}^{p-1} = M^{p-1}, \Theta_{<p} = 0, \Theta_p = 1, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right] \\
&= \mathbb{E}\left[\mathcal{S} \mid \Theta_p = 1\right] && \text{(change of notation)} \\
&= \sum_s s \cdot \Pr\left(\mathcal{S} = s \mid \Theta_p = 1\right) \\
&\geqslant \sum_s s \cdot \frac{1}{2} \cdot \Pr\left(\mathcal{S} = s\right) \\
&= \frac{1}{2} \cdot \mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \Pi^{1:p-1} = \Pi^{1:p-1}, \mathsf{M}^{p-1} = M^{p-1}, \Theta_{<p} = 0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right],
\end{aligned}$$

as desired. ■

Proof of Lemma 5 We first lower bound the total expected number of arm pulls with the expected number of arm pulls *restricting to* the first $P-p+2$ batches, and write the expectation as the average case of the choices of $(\Pi^{1:p-1}, M^{p-1})$.

$$\begin{aligned}
&\mathbb{E}\left[\mathsf{Smp}(\text{ALG}) \mid \mathcal{E}_{\text{First}}(p), \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right] \\
&= \mathbb{E}_{\Pi^{1:p-1},\mathsf{M}^{p-1}} \mathbb{E}\left[\mathsf{Smp}(\text{ALG}) \mid \Pi^{1:p-1}, M^{p-1}, \mathcal{E}_{\text{First}}(p), \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right] \\
&\geqslant \mathbb{E}_{\Pi^{1:p-1},\mathsf{M}^{p-1}} \mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \Pi^{1:p-1}, M^{p-1}, \mathcal{E}_{\text{First}}(p), \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right],
\end{aligned}$$

where the last inequality is due to $\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}}$ always counts a subset of arm pulls of $\mathsf{Smp}(\text{ALG})$ for any fixed transcript and memory $(\Pi^{1:p-1}, M^{p-1})$. By applying Lemma 18 to *every* choice of batch- and memory-oblivious $(\Pi^{1:p-1}, M^{p-1})$, we have

$$\begin{aligned}
&\mathbb{E}_{\Pi^{1:p-1},\mathsf{M}^{p-1}} \mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \Pi^{1:p-1}, M^{p-1}, \Theta_{<p}=0, \Theta_p=1, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right] \\
&\geqslant \frac{1}{2} \cdot \mathbb{E}_{\Pi^{1:p-1},\mathsf{M}^{p-1}} \mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \Pi^{1:p-1}, M^{p-1}, \Theta_{<p}=0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right].
\end{aligned}$$

To avoid clutter, we can combine the above inequalities and re-write them in the form of

$$\begin{aligned}
&\mathbb{E}\left[\mathsf{Smp}(\text{ALG}) \mid \mathcal{E}_{\text{First}}(p), \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right] \\
&\geqslant \frac{1}{2} \cdot \mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \Theta_{<p}=0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right].
\end{aligned}$$

Therefore, by the assumption of Lemma 5, we have

$$\begin{aligned}
&\mathbb{E}\left[\mathsf{Smp}(\text{ALG}) \mid \mathcal{E}_{\text{First}}(p), \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right] \\
&\geqslant \mathbb{E}\left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \Theta_{<p}=0, \Theta_p=1, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right] \\
&\geqslant \frac{1}{2} \cdot \left[\mathsf{Smp}(\text{ALG})_{B_{P+1}:B_{p+1}} \mid \Theta_{<p}=0, \mathcal{E}^{p-1}_{\text{batch-obl}}, \mathcal{E}^{p-1}_{\text{mem-obl}}\right] && \text{(by Lemma 18)} \\
&> \frac{1}{10^{10}} \cdot \frac{n}{\gamma_{p+1} \cdot P^{30}} && \text{(by the condition of Lemma 5)} \\
&= \frac{1}{10^{10}} \cdot \frac{n}{\eta_p^2 \cdot P^{30}} \cdot (6C \cdot P)^{30} && \text{(by the construction } \eta_{p+1} = \left(\tfrac{1}{6CP}\right)^{15} \cdot \eta_p) \\
&> 20000C \cdot \frac{n}{\eta_p^2}, && (6^{30}/10^{10} > 20000 \text{ with } C \geqslant 1)
\end{aligned}$$

as desired by Lemma 5. ■

D.3. Putting Everything Together: Proof of Theorem 1

We now prove Theorem 1 with Lemmas 4 and 5. We remind the readers that we use ALG to denote the streaming algorithm. Note that in the beginning of the first pass, ALG is necessarily memory- and batch-oblivious since there is $\Pi^0 = \emptyset$ and $M^0 = \emptyset$. Therefore, by Lemma 5, if the algorithm enters the *radical case*, there is

$$\mathbb{E}\left[\mathsf{Smp}(\text{ALG}) \mid \mathcal{E}_{\text{First}}(1)\right] = \mathbb{E}\left[\mathsf{Smp}(\text{ALG}) \mid \Theta_1 = 1\right] > C \cdot \frac{n}{\eta_1^2},$$

which breaks the sample complexity requirement in Theorem 1. Therefore, ALG must use the *conservative case* for the first pass.

Starting from the second pass, we argue that no pass should use the *radical case* if ALG is to follow the upper bound on the sample complexity as required by Theorem 1. Suppose $\tilde{p}$ is the first pass that the algorithm enters the *radical case*, and since we have the base case of $p = 1$ and the condition of Lemma 4 (conservative case) being satisfied before pass $\tilde{p}$, there is

$$\Pr\left(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}}, \Theta_{<\tilde{p}} = 0\right) \geqslant \left(1 - \frac{1}{2P}\right)^{10(\tilde{p}-1)}.$$

We use the above result to lower bound the probability for $\Pr(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}} \mid \mathcal{E}_{\text{First}}(\tilde{p}))$, which will eventually lead to a lower bound on $\mathbb{E}\left[\mathsf{Smp}(\text{ALG}) \mid \mathcal{E}_{\text{First}}(\tilde{p})\right]$ that breaks the limit of samples.

To this end, we first show the following technical claim that allows us to "drop" conditions on $\Theta_{\tilde{p}}$ conditioning on $\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}$ and $\mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}}$. Intuitively, such a claim is true by the obliviousness of the transcipt on $\Theta_{\tilde{p}}$, which is similar-in-spirit with Lemma 18.

Claim D.2 *The following statement is true:*

$$\Pr\left(\mathcal{E}^{\tilde{p}-1}_{batch\text{-}obl}, \mathcal{E}^{\tilde{p}-1}_{mem\text{-}obl} \mid \mathcal{E}_{\text{First}}(\tilde{p})\right) \geqslant \frac{1}{2} \cdot \Pr\left(\mathcal{E}^{\tilde{p}-1}_{batch\text{-}obl}, \mathcal{E}^{\tilde{p}-1}_{mem\text{-}obl}, \Theta_{<\tilde{p}} = 0\right).$$

Proof We first lower bound the probability by expanding the terms as follows.

$$\begin{aligned}
\Pr\left(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}} \mid \mathcal{E}_{\text{First}}(\tilde{p})\right) &= \frac{\Pr\left(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}}, \mathcal{E}_{\text{First}}(\tilde{p})\right)}{\Pr\left(\mathcal{E}_{\text{First}}(\tilde{p})\right)} \\
&\geqslant \Pr\left(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}}, \mathcal{E}_{\text{First}}(\tilde{p})\right) \cdot 2P \\
&\qquad\qquad (\Pr\left(\mathcal{E}_{\text{First}}(\tilde{p})\right) \leqslant 1/2P) \\
&= \Pr\left(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}}, \Theta_{\tilde{p}} = 1, \Theta_{<\tilde{p}} = 0\right) \cdot 2P \\
&= \Pr\left(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}}, \Theta_{<\tilde{p}} = 0 \mid \Theta_{\tilde{p}} = 1\right) \cdot \Pr\left(\Theta_{\tilde{p}} = 1\right) \cdot 2P \\
&= \Pr\left(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}}, \Theta_{<\tilde{p}} = 0 \mid \Theta_{\tilde{p}} = 1\right). \\
&\qquad\qquad (\Pr(\Theta_{\tilde{p}} = 1) = \tfrac{1}{2P} \text{ by } \tilde{p} \leqslant p)
\end{aligned}$$

Our goal now is to lower bound the term $\Pr\left(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}}, \Theta_{<\tilde{p}} = 0 \mid \Theta_{\tilde{p}} = 1\right)$. By Bayes' rule and the batch-obliviousness, we have

$$\begin{aligned}
&\Pr\left(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}}, \Theta_{<\tilde{p}} = 0 \mid \Theta_{\tilde{p}} = 1\right) \\
&= \frac{\Pr\left(\Theta_{\tilde{p}} = 1 \mid \mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}}, \Theta_{<\tilde{p}} = 0\right) \cdot \Pr\left(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}}, \Theta_{<\tilde{p}} = 0\right)}{\Pr\left(\Theta_{\tilde{p}} = 1\right)} && \text{(Bayes' rule)} \\
&\geqslant \frac{\left(\frac{1}{4P}\right) \cdot \Pr\left(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}} \mid \Theta_{<\tilde{p}} = 0\right)}{\frac{1}{2P}} && \text{(by the batch-oblivious condition)} \\
&\geqslant \frac{1}{2} \cdot \Pr\left(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}}, \Theta_{<\tilde{p}} = 0\right),
\end{aligned}$$

as desired. ■

We now establish the lower bound on the expected sample for pass $\tilde{p}$. By Claim D.2, we have that

$$\begin{aligned}
\Pr\left(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}} \mid \mathcal{E}_{\text{First}}(\tilde{p})\right) &\geqslant \frac{1}{2} \cdot \left(1 - \frac{1}{2P}\right)^{10(\tilde{p}-1)} \\
&\geqslant \frac{1}{2} \cdot \left(1 - \frac{1}{2P}\right)^{10(P-1)} \\
&> \frac{1}{1000},
\end{aligned} \tag{7}$$

where the first inequality uses Claim D.2 and the lower bound on $\Pr(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}}, \Theta_{<\tilde{p}} = 0)$, and the last inequality is obtained by using $\left(1 - \frac{1}{2P}\right)^{10P-10} > \frac{1}{500}$ for any $P \geqslant 2$. Therefore, we can bound the sample complexity of the algorithm if it enters the *radical case* on the $\tilde{p}$-th pass as follows.

$$\begin{aligned}
&\mathbb{E}\left[\mathsf{Smp}(\textsc{alg}) \mid \mathcal{E}_{\text{First}}(\tilde{p})\right] \\
&\geqslant \mathbb{E}\left[\mathsf{Smp}(\textsc{alg}) \mid \mathcal{E}_{\text{First}}(\tilde{p}),\, \mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}},\, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}}\right] \\
&\qquad \cdot \Pr\left(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}},\, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}} \mid \mathcal{E}_{\text{First}}(\tilde{p})\right) \\
&> 20000C \cdot \frac{n}{\eta_p^2} \cdot \frac{1}{1000} \\
&\qquad \text{(by Lemma 5 and the lower bound of } \Pr\left(\mathcal{E}^{\tilde{p}-1}_{\text{batch-obl}}, \mathcal{E}^{\tilde{p}-1}_{\text{mem-obl}} \mid \mathcal{E}_{\text{First}}(\tilde{p})\right)) \\
&> C \cdot \frac{n}{\eta_p^2},
\end{aligned}$$

which breaks the requirement of sample complexity bound in Theorem 1. As such, to keep the promise on the sample complexity, ALG has to be in the *conservative case* for *all* P passes.

Now, we can apply the calculation in Eq (7) again to argue that with probability strictly more than $\frac{1}{1000}$, after the P-th pass, we obtain transcript and memory that are memory- and batch-oblivious. As such, no arm with a mean reward strictly more than $1/2$ will be in the memory of ALG, which means the success probability is strictly less than $\frac{999}{1000}$.

Remark 19 *Observe that our lower bound generalizes to a sample complexity with additional* $\text{polylog}\,(\frac{1}{\eta_p})$ *multiplicative factors, i.e., we can prove the lower bound of memory and success probability in Theorem 1 with the condition of*

$$\mathbb{E}\left[\mathsf{Smp}(\text{ALG}) \mid \mathcal{E}_{\text{First}}(p)\right] \leqslant C \cdot \frac{n}{\eta_p^2} \cdot \text{polylog}\,(\frac{1}{\eta_p}).$$

In particular, if we further increase the gap between η_p *in different batches, e.g., if we use* $\eta_p = \left(\frac{1}{6\,C\cdot P}\right)^{20p}$ *in Eq (1), we can bring an extra multiplicative term of* $\text{poly}(P)$ *to the sample complexity in our proof of Lemma 5. By our choice of parameters, we have* $P = \Omega(\frac{\log(1/\widetilde{\Delta})}{\log\log(1/\widetilde{\Delta})})$, *where* $\widetilde{\Delta} \leqslant \eta_p$ *for any* $p \in [P+1]$. *This leads to the desired bound on sample complexity. The observation also strengthens our main lower bound result to an expected sample complexity of* $O(\frac{n}{\Delta^2} \cdot \text{polylog}\,(\frac{1}{\Delta}))$.