Nearly Tight Bounds for Exploration in Streaming Multi-Armed Bandits with Known Optimality Gap

Nikolai Karpov[1], **Chen Wang**[2,3]
[1]Indiana University
[2] Rice University
[3] Texas A&M University
kimaska@gmail.com, cwangjhw@tamu.edu

Abstract

We investigate the sample-memory-pass trade-offs for pure exploration in multi-pass streaming multi-armed bandits (MABs) with the *a priori* knowledge of the optimality gap $\Delta_{[2]}$. Here, and throughout, the optimality gap $\Delta_{[i]}$ is defined as the mean reward gap between the best and the i-th best arms. A recent line of results have shown that if there is no known $\Delta_{[2]}$, a pass complexity of $\tilde{\Theta}(\log(1/\Delta_{[2]}))$ is necessary and sufficient to obtain the *worst-case optimal* $O(n/\Delta^2_{[2]})$ sample complexity with a single-arm memory. However, our understanding of multi-pass algorithms with known $\Delta_{[2]}$ is still limited. The key open problem is how many passes are required to achieve the complexity, i.e., $O(\sum_{i=2}^n 1/\Delta^2_{[i]} \cdot \log n)$ arm pulls, with a sublinear memory size.

In this work, we show that the "right answer" for the question is $\tilde{\Theta}(\log n)$ passes. We first present a lower bound, showing that any algorithm that finds the best arm with slightly sublinear memory – a memory of $o(n/\text{polylog}(n))$ arms – and $O(\sum_{i=2}^n 1/\Delta^2_{[i]} \cdot \log n)$ arm pulls has to make $\Omega(\frac{\log n}{\log\log n})$ passes over the stream. We then show a nearly-matching algorithm that assuming the knowledge of $\Delta_{[2]}$, finds the best arm with $O(\sum_{i=2}^n 1/\Delta^2_{[i]} \cdot \log n)$ arm pulls and a *single arm* memory.

1 Introduction

The pure exploration multi-armed bandits (MABs) is one of the most well-studied problems in theoretical computer science (TCS) and machine learning (ML). The problem is formulated as follows: given n arms with unknown sub-Gaussian reward distributions, find the best arm, defined as the arm with the highest mean reward, with a sufficiently high probability and a small number of arm pulls (sample complexity). Here, and throughout, the parameter $\Delta_{[i]}$ is defined as the mean reward gap between the best and the i-th best arms. The problem has been extensively studied in the literature (e.g., (Even-Dar, Mannor, and Mansour 2002; Mannor and Tsitsiklis 2004; Kalyanakrishnan and Stone 2010; Karnin, Koren, and Somekh 2013; Jamieson et al. 2014; Kaufmann, Cappé, and Garivier 2016; Carpentier and Locatelli 2016; Agarwal et al. 2017; Chen, Li, and Qiao 2017a), see (Slivkins 2019) for an excellent monograph), and its application has been found in numerous areas, e.g., experiment design (Robbins 1952; Villar, Bowden, and Wason 2015; Aziz, Kaufmann, and Riviere 2021), recommendation systems (Silva et al. 2022), search ranking (Agarwal et al. 2008; Radlinski, Kleinberg, and Joachims 2008), robot control (Koval et al. 2015), to name a few.

The optimal sample complexity bound under the classical RAM setting has been established through a series of elegant works (Even-Dar, Mannor, and Mansour 2002; Mannor and Tsitsiklis 2004; Karnin, Koren, and Somekh 2013; Jamieson et al. 2014). The pioneering work of (Even-Dar, Mannor, and Mansour 2002) shows that if the value of $\Delta_{[2]}$ is known, there exists an algorithm that finds the best arm with high (constant) probability and a *worst-case optimal* $O(n/\Delta^2_{[2]})$ arm pulls. Subsequently, the work of (Karnin, Koren, and Somekh 2013; Jamieson et al. 2014) improved the sample complexity to the *nearly instance optimal* bound of $O(\sum_{i=2}^n 1/\Delta^2_{[i]} \cdot \log\log(1/\Delta_{[i]}))$[1], and their algorithms do *not* require a known value of $\Delta_{[2]}$. On the lower bound side, (Mannor and Tsitsiklis 2004) showed that a sample complexity of $\Omega(\sum_{i=2}^n 1/\Delta^2_{[i]})$ is necessary to find the best arm with high constant probability, which completes the picture for pure exploration under the RAM model up to the doubly-logarithmic factor.[2]

Virtually all algorithms for pure exploration in the classical setting require all arms available in the memory for repeated visits. Aimed at modern large-scale applications, in which storing everything becomes impossible, an important direction to explore is MABs in the memory-constrained setting. To this end, (Assadi and Wang 2020) introduced the streaming MABs model, where the arms arrive one by one in a streaming manner, and the algorithm uses a limited memory to store, discard, and replace arms. The target for streaming MABs algorithms is to simultaneously minimize the sample complexity and the *space complexity* – the maximum number of arms stored at any point. (Assadi and Wang 2020) showed

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1]We slightly overload the terms to call both $O(\sum_{i=2}^n 1/\Delta^2_{[i]} \cdot \log\log(1/\Delta_{[i]}))$ and $O(\sum_{i=2}^n 1/\Delta^2_{[i]} \cdot \text{polylog}(n))$ nearly instance optimal sample complexity, and denote (any of) them with $\tilde{O}(\sum_{i=2}^n 1/\Delta^2_{[i]})$ when the context is clear.

[2]It turns out the seemingly artificial $\log\log(1/\Delta_{[i]})$ factor embodies some fundamental properties of the problem. See (Chen and Li 2016; Chen, Li, and Qiao 2017b) for more discussions.

that if the value of $\Delta_{[2]}$ is provided, there exists a *single-pass* algorithm that finds the best arm with high constant probability, the worst-case optimal $O(n/\Delta_{[2]}^2)$ sample complexity, and a *single-arm* memory. The key conceptual message of (Assadi and Wang 2020) is that in the regime where $\Delta_{[2]}$ is known and the target is the worst-case optimal sample complexity, there is no sample-space trade-off in this setting.

The results of (Assadi and Wang 2020) have led to considerable interest in understanding the power and limitations of the streaming MABs model (Maiti, Patil, and Khan 2021; Jin et al. 2021; Assadi and Wang 2022; Agarwal, Khanna, and Patil 2022; Wang 2023; Li et al. 2023). In particular, since (Assadi and Wang 2020) only deals with the setting when $\Delta_{[2]}$ is given and the worst-case optimal bound, a natural question is to ask what happens if $\Delta_{[2]}$ is unknown, or if the target becomes the (nearly) instance optimal bound instead. Unfortunately, in these settings, the optimistic message in (Assadi and Wang 2020) no longer holds: (Assadi and Wang 2022) showed that in the single-pass setting, if the value of $\Delta_{[2]}$ is not given a priori, then the sample complexity is unbounded unless the algorithm has $\Omega(n)$-arm memory. Furthermore, even if the value of $\Delta_{[2]}$ is known, there is a sample complexity lower bound of $\Omega(n/\Delta_{[2]}^2)$ for any algorithm with $o(n)$-arm memory in the single-pass streaming setting. These results assert that multiple passes over the stream are necessary if we want any streaming algorithms with sublinear memory under the new settings.

It turns out that allowing multiple passes does lead to improved bounds. Concretely, (Jin et al. 2021) shows that in $O(\log(1/\Delta_{[2]}))$ passes, it is possible to find the best arm with a single-arm memory and the near-instance optimal $O(\sum_{i=2}^n 1/\Delta_{[i]}^2 \cdot \log\log(1/\Delta_{[i]}))$ arm pulls. Furthermore, the algorithm does *not* require a known quantity of $\Delta_{[2]}$. Very recently, (Assadi and Wang 2024) proved that for a streaming algorithm with $o(n)$-arm memory to achieve even the worst-case optimal bound $O(n/\Delta_{[2]}^2)$ bound with the stream alone, a number of $\Omega\left(\frac{\log(1/\Delta_{[2]})}{\log\log(1/\Delta_{[2]})}\right)$ passes is necessary. As such, we already established a good understanding of the pass-sample-space trade-off for multi-pass algorithms without $\Delta_{[2]}$ value.

The final missing piece to complete the theoretical picture of multi-pass streaming MABS is to understand the case when $\Delta_{[2]}$ is provided *and* the target is the instance-optimal sample complexity. We remark this question is not trivial: in the adversarial instance distribution of (Assadi and Wang 2024), if $\Delta_{[2]}$ is provided, we can uniquely determine the realization of instances in their distribution. As such, it is possible that if $\Delta_{[2]}$ is known, there are algorithms with better efficiency. This open question can be formally presented as follows.

If the value of $\Delta_{[2]}$ is known a priori*, what is the optimal number of passes for a streaming algorithm with $o(n)$ arm memory to find the best arm with the (nearly) instance optimal sample complexity?*

We now provide some additional discussions to better motivate the investigation. The importance of the open question could be summarized as follows.

- The question is important for the theoretical foundations of *online learning*. The streaming MABs model has been widely regarded as an important models for modern large-scale online learning (Maiti, Patil, and Khan 2021; Jin et al. 2021; Assadi and Wang 2022; Agarwal, Khanna, and Patil 2022; Wang 2023; Li et al. 2023; Assadi and Wang 2024). Since the knowledge of $\Delta_{[2]}$ is frequently assumed in the literature, as evidenced by works such as (Even-Dar, Mannor, and Mansour 2002; Assadi and Wang 2020), the motivating question is an important missing piece to be answered for multi-pass pure exploration. As such, the primary motivation for the investigation is to complete the theoretical picture for the streaming MABs model.
- Algorithms with better sample and pass efficiency could lead to *direct application* in various tasks. For instance, if we want to find the best seller in large-scale online retailers, we could query the data from data centers in a streaming fashion, and run the algorithm using the local RAM. Here, we could *estimate* the value of $\Delta_{[2]}$ from historical data. And since the products often have very different scales of transactions, which implies that $\frac{n}{\Delta_{[2]}^2} \gg \sum_{i=2}^n \frac{1}{\Delta_{[i]}^2}$, our algorithm is much more efficient than the algorithm of (Assadi and Wang 2020; Jin et al. 2021).

Our Contributions

Our main contribution is the answer to open question: we provide nearly-matching (up to exponentially smaller terms) upper and lower bounds for streaming algorithms with $o(n)$-arm memory to find the best arm with a (nearly) instance optimal sample complexity. We first present our lower bound result as follows.

Result 1 (Lower bound, informal of Theorem 1)**.** Any streaming algorithm that given n arms in a stream and a known value of $\Delta_{[2]}$, finds the best arm with high constant probability, $O(\sum_{i=2}^n \frac{1}{\Delta_{[i]}^2} \cdot \log(n))$ arm pulls, and $o(\frac{n}{\text{polylog}(n)})$ arm memory has to use $\Omega(\frac{\log(n)}{\log\log(n)})$ passes over the stream.

Result 1 shares a similar form of the (Assadi and Wang 2024), albeit we are able to substitute $\log(1/\Delta_{[i]})$ terms with $\log n$ terms. Before our results, the only known result for streaming MABs lower bounds with instance-optimal sample complexity is th result of (Assadi and Wang 2022), which only works for a *single* pass. Therefore, Result 1 marks a significant improvement in the pass complexity of the problem.

A natural question to follow up Result 1 is to answer whether the lower is optimal, i.e., is it possible to design an algorithm with a sample and pass complexity that matches the lower bound in Result 1. We answer this question in the affirmative by showing an algorithm as in Result 2.

Result 2 (Upper bound, informal of Theorem 2)**.** There exists a streaming algorithm that given n arms in a stream and the value of $\Delta_{[2]}$, finds the best arm with high

constant probability with $O(\sum_{i=2}^{n} \frac{1}{\Delta_{[i]}^2} \cdot \log(n))$ arm pulls, $O(\log(n))$ passes over the stream, and a memory of a single arm.

In fact, the guarantee of our algorithm extends beyond just $O(\log(n))$ passes. We can always keep using only a single arm memory, and set P passes to achieve $O(\sum_{i=2}^{n} \frac{1}{\Delta_{[i]}^2} \cdot n^{2/P} \cdot \log(nP))$ sample complexity. The sample-pass trade-off is optimized by taking $P = O(\log(n))$. The sample complexity bound of $O(\sum_{i=2}^{n} \frac{1}{\Delta_{[i]}^2} \cdot \log(n))$ we obtain here is slightly different from the classical near-instance optimal bound of $O(\sum_{i=2}^{n} \frac{1}{\Delta_{[i]}^2} \cdot \log\log(1/\Delta_{[i]}))$. We remark that the $\log(n)$ multiplicative factor does *not* render the bound trivial.

Result 2 suggests that algorithms with known $\Delta_{[2]}$ values could have much better pass efficiency. Note that the lower bound in (Assadi and Wang 2024) holds with $\Delta_{[2]}$ values as small as $2^{n^{O(1)}}$, which means we may be forced to take poly(n) passes if $\Delta_{[2]}$ is unknown. In contrast, in the case when $\Delta_{[2]}$ is known, it becomes possible to find the best arm in $\log(n)$ passes, which is much smaller and reasonable in practice.

To make it easier for the readers to understand the context and contributions of our results, we provide Table 1 that illustrates the comparison between the existing results and ours.

Result 1 and Result 2 demonstrate a sharp memory-pass trade-off for streaming algorithms to find the best arm with a near instance optimal sample complexity: with $O(\log(n))$ passes, we can obtain an algorithm with a memory of a single arm. However, if we decrease the number of passes slightly to $o(\log(n)/\log\log(n))$, no streaming algorithm will be able to achieve the sample complexity and success probability guarantee unless it uses almost n-arm memory. We note that this kind of dichotomy frequently arises in the streaming MABs literature ((Assadi and Wang 2020, 2022; Agarwal, Khanna, and Patil 2022; Assadi and Wang 2024)), and we obtain a similar phenomenon in the multi-pass setting with a known $\Delta_{[2]}$ as well (see Table 1 for some examples).

Finally, we observe that by simply running our streaming algorithm in the offline setting, we obtain an offline algorithm that finds the best arm with high constant probability and $O(\sum_{i=2}^{n} \frac{1}{\Delta_{[i]}^2} \cdot \log(n))$ arm pulls. This observation, while straightforward, has the following implication on the role of $\Delta_{[2]}$ in the pure exploration MABs problem. The classical *nearly* instance optimal sample complexity bound by (Karnin, Koren, and Somekh 2013; Jamieson et al. 2014) is $O(\sum_{i=2}^{n} \frac{1}{\Delta_{[i]}^2} \cdot \log\log(\frac{1}{\Delta_{[i]}}))$. In the two-arm scenario, (Jamieson et al. 2014) also proved that $\Omega(\frac{1}{\Delta_{[2]}^2} \cdot \log\log(\frac{1}{\Delta_{[2]}}))$ arm pulls are *necessary* (when the value of $\Delta_{[2]}$ is unknown). (Chen and Li 2015) further improved the upper bound to $O(\sum_{i=2}^{n} \frac{1}{\Delta_{[i]}^2} \cdot \log\log(\min\{n, \frac{1}{\Delta_{[i]}}\}) + \frac{1}{\Delta_{[2]}^2} \cdot \log\log(\frac{1}{\Delta_{[2]}}))$, and the discussion went deeper with later results by (Chen and Li 2016; Chen, Li, and Qiao 2017b), in which they proposed the 'gap-entropy conjecture' for the optimal sample complexity bound (also for the case of unknown $\Delta_{[2]}$, see (Chen and Li 2016) for details). In contrast, our observation shows that if $\Delta_{[2]}$ is provided, we can instead get a multiplicative term that is independent of all $\Delta_{[i]}$ values in the logarithmic term. This observation might be of independent interests in the broader MABs community.

Experiments. To validate the performance of our algorithm, we conduct experiments on multiple types of streaming MABs instances. We compare our algorithm with two benchmarks: i). the single-pass algorithm by (Assadi and Wang 2020), which enjoys the ultimate pass efficiency of a single pass, but only guarantees the worst-case optimal $\Theta(\frac{n}{\Delta_{[2]}^2})$ sample complexity; and ii). the $O(\log(1/\Delta_{[2]}))$-pass algorithm by (Jin et al. 2021), which has the advantage of not requiring the knowledge of $\Delta_{[2]}$, but has to use more passes. Our result shows that in multiple setting, our algorithm consistently enjoys the best sample efficiency. Furthermore, comparing ot the algorithm of (Jin et al. 2021), our algorithm uses significantly less passes over the stream. The results of our experiments are presented in Section 5.

Our Techniques

Lower Bound. Proving lower bounds for multi-pass streaming MABs typically involves intricate techniques to carefully manage memory, samples, and the information revealed over time. To navigate these technical challenges, we draw inspiration from recent work by (Assadi and Wang 2024), which established lower bounds for multi-pass MABs without prior knowledge of $\Delta_{[2]}$. The lower bound construction devised by (Assadi and Wang 2024) employs a 'batched' approach to distributing instances. Roughly, it divides the arms into $B + 1$ batches for a P-pass algorithm, where $P \leq B$. Within each batch b, all but one arm consistently yield a mean reward of $\frac{1}{2}$, while the remaining 'special arm' offers stochastic mean rewards –either $\frac{1}{2}$ or $\frac{1}{2} + \alpha_b$ for some $\alpha_b > 0$– placed uniformly at random among the arms within batch b. Importantly, the later-arriving batches *might* have higher mean rewards. This construction allows us to argue that to maintain optimal sample complexity, the algorithm must always check whether a batch contains an arm with a reward exceeding $1/2$ from the *latest* batch that has not been checked, which forms a lower bound.

The above sketches the *intuition* of (Assadi and Wang 2024), and the actual proof is considerably more involved. Despite the very technical analysis, we observe that we could actually extract a framework from (Assadi and Wang 2024) to capture the memory-sample trade-off for algorithms on batched instances. In particular, to establish such trade-offs, we only need i). a sample complexity lower bound for the streaming algorithm to 'trap' the special arm from a batch; ii). a sample lower bound for the streaming algorithm to 'learn' the distribution for each batch; and iii). a sufficiently high gap between the sample complexity for different batches. We remark that these aspects are not explicitly written in (Assadi and Wang 2024), and forming the framework from key observations is one of our technical contributions.

With this novel technical framework, a natural idea to prove lower bounds for the instance-sensitive

Pass	$\Delta_{[2]}$ is given	Sample Complexity	Memory	Remark and Reference
1	Yes	$O(\frac{n}{\Delta_{[2]}^2})$	1	Upper bound, (Assadi and Wang 2020)
1	No	$O(\frac{n}{\Delta_{[2]}^2})$	$\Omega(n)$	Lower bound, (Assadi and Wang 2022)
1	Yes	$\tilde{O}(\sum_{i=2}^{n} \frac{1}{\Delta_{[i]}^2})$	$\Omega(n)$	Lower bound, (Assadi and Wang 2022)
$O(\log(\frac{1}{\Delta_{[2]}}))$	No	$\tilde{O}(\sum_{i=2}^{n} \frac{1}{\Delta_{[i]}^2})$	1	Upper bound, (Jin et al. 2021)
$O(\frac{\log(\frac{1}{\Delta_{[2]}})}{\log\log(\frac{1}{\Delta_{[2]}})})$	No	$O(\frac{n}{\Delta_{[2]}^2})$	$\Omega(n/\text{polylog}(\frac{1}{\Delta_{[2]}}))$	Lower bound, (Assadi and Wang 2024)
$O(\frac{\log(n)}{\log\log(n)})$	Yes	$\tilde{O}(\sum_{i=2}^{n} \frac{1}{\Delta_{[i]}^2})$	$\Omega(n/\text{polylog}(n))$	Lower bound, Result 1
$O(\log n)$	Yes	$\tilde{O}(\sum_{i=2}^{n} \frac{1}{\Delta_{[i]}^2})$	1	Upper bound, Result 2

Table 1: Summary of the previous results and our new results. To present the sample-memory-pass trade-offs, we set upper bounds on the number of passes and sample complexity, and show the memory in terms of the number of arms with both upper and lower bounds.

$O(\sum_{i=2}^{n} 1/\Delta_{[i]}^2 \cdot \log(n))$ sample complexity is to construct a batched instance distribution that a). satisfies the properties as required by the framework, and b). varies the quantity of $O(\sum_{i=2}^{n} 1/\Delta_{[i]}^2 \cdot \log(n))$ with different realizations. To this end, we note that we could *not* directly use the construction of (Assadi and Wang 2024) in our setting. An obvious problem here is that since the values of α_b change across batches, the information of $\Delta_{[2]}$ alone can help uniquely identify the realization of the distribution, which renders the distribution not hard. The idea to resolve this issue is to have *two* special arms, whose mean rewards are with $\frac{1}{2} + \chi_b$ and $\frac{1}{2} + \chi_b + \gamma$. Moreover, we make χ_b to be much larger than γ for any b, but progressively smaller by $1/\text{poly}(B)$ factor across the batches. As such, we always have $\Delta_{[2]} = \gamma$, which means the value of $\Delta_{[2]}$ no longer reveals any information about the instance realization.

The final missing piece is to show how 'hard' the new construction is for streaming algorithms. Unfortunately, due to the introduction of an extra special arm, the standard technical tools developed from (Agarwal, Khanna, and Patil 2022; Assadi and Wang 2022, 2024) no longer work. To overcome the issue, we use the information-theoretic tools to develop several new results for *double-armed bandits*, and apply the 'direct-sum' idea in (Assadi and Wang 2022, 2024) to obtain new sample complexity bounds for batches with two special arms. To the best of our knowledge, this is the first lower bound that studies the sample complexity in the *double-armed* setting, which could be of independent interest. Finally, by taking $B = \Theta(\log(n)/\log\log(n))$, we can simultaneously ensure the value gap between χ_b and γ and the sample complexity gap between batches, which gives the desired result.

Upper Bound. Our algorithms work with the elimination-based approach extensively studied in the MABs literature (Hillel et al. 2013; Karpov, Zhang, and Zhou 2020). The state-of-the-art multi-pass streaming algorithm by (Jin et al. 2021) is a streaming adaptation of the classical elimination algorithm (Karnin, Koren, and Somekh 2013). The algorithm by (Jin et al. 2021) requires $O\left(\log\left(\frac{1}{\Delta_{[2]}}\right)\right)$ passes, and the main idea is to "binary search" the "correct" gap parameters. Concretely, at the beginning of the p-th pass, the algorithm fixes an elimination gap $\epsilon_p = O(2^{-p})$, the goal of the algorithm at the end of the p-pass is to eliminate all arms i such that $\Delta_i > \epsilon_p$[3] with roughly $O(1/\epsilon_p^2)$ arm pulls on arm i. After $O\left(\log\left(\frac{1}{\Delta_{[2]}}\right)\right)$ passes, the value of the elimination gap becomes smaller than $\Delta_{[2]}$, and all arms except the best arm can be safely eliminated.

To make the number of passes independent of $\Delta_{[2]}$, our key observation is that the "binary search" in the elimination procedure can be made more efficient with geometric series from $n\Delta_{[2]}$ to $\Delta_{[2]}$. Concretely, instead of initiating the elimination process of arms with constant reward gap, we choose the sequence of elimination gaps in the form of $\epsilon_p = \Delta_{[2]} \cdot n^{1-p/P}$ for the p-th pass. Here, P is the total number of passes for the algorithm. On a very high level, it is easy to observe that after P passes, the elimination gaps become smaller than $\Delta_{[2]}$, and all arms except the best arm can be safely eliminated (with high probability). The observation generalizes to any arm_i with gap parameter Δ_i: when ϵ_p become smaller than Δ_i, a sup-optimal arm arm_i is eliminated with high probability. The correctness of the algorithm thus follows from the high probability event that only the best arm will remain after P passes.

For the analysis of sample complexity, we proceed by categorizing the set of arms into two parts with large gaps $\Delta_i > n\Delta_{[2]}$ and small gaps $\Delta_i \leq n\Delta_{[2]}$. The analysis for a small gaps group controls that the number of pulls assigned to each arm is at most $O(\frac{n^{1/P}}{\Delta_i^2})$. Similar to the analysis of (Karnin, Koren, and Somekh 2013; Jin et al. 2021), the key observation here is that after the number of pulls used on arm i becomes larger than $\frac{1}{\Delta_i^2}$, we can eliminate such a suboptimal arm, which implies that we spend at most $\frac{n^{2/P}}{\Delta_i^2}$ pulls for arm

[3]We use the notation Δ_i (without the brackets on i) to denote the gap between the best arm and arm_i. See Section 2 for more clarifications.

i with high probability. On the other hand, for the arms with large gaps, we observe that the sample complexity of *all* arms with gaps more than $n\Delta_{[2]}$ is dominated by a single largest term $\frac{1}{\Delta_{[2]}^2}$, which leads to the desired sample complexity bound.

2 Preliminary

We introduce the notation and formal description of the streaming MABs model in this section. We provide more technical preliminaries in the appendix.

Notation

Throughout, we use n to denote the number of arms. We use i to denote the indices of the arms, and we have the set of indices as I (which is a permutation of $[n]$). We let μ_i be the mean of the i-th arm; furthermore, we denote the index of the best arm as $\star := \arg\max_{i \in I} \mu_i$. As such, the best arm is denoted as arm*, and the mean of the best arm is $\mu_\star$. The reward gap between the best and the i-th arm is equal to $\Delta_i := \mu_\star - \mu_i$. We also use the ordered sequence of gaps $\Delta_{[2]} \leq \Delta_{[3]} \leq \ldots \leq \Delta_{[n]}$, i.e., $\Delta_{[i]}$ is the reward gap between the best and the i-th *best* arm.

We frequently deal with Bernoulli random variables. For convenience, we use Bern(μ) to denote a Bernoulli distribution with mean μ (i.e., the probability to sample 1 is μ). When random variables and their realizations are presented side by side, as a convention, we use upper cases (e.g., Π) to denote the random variables and lower cases (e.g., π) to denote the realizations.

The Streaming Multi-armed Bandits Model

We now formally introduce the streaming MABs model as follows. There is a collection of n arms, denoted as $\{\text{arm}_i\}_{i=1}^n$, and their reward distributions are characterized by $\{\text{Bern}(\mu_i)\}_{i=1}^n$[4]. As the name suggested, the arms arrive one after another in an *arbitrary and fixed* order (a permutation over $[n]$). Here, an *arbitrary* order means the arrival order of the arms is selected by an adversary and can be in the worst case, and a *fixed* order means that the order of arrival for the arms is the same across different passes.

A multi-pass streaming algorithm in the streaming MABs setting is defined as an algorithm that maintains a memory M, which is a set of arms, and a transcript π, which encodes the statistics of all past arm pulls. Each record in π is a tuple that specifies the identity of the pulled arm, the result, and the pass index when the sample happened.

At any point, the streaming algorithm is allowed to make an arbitrary number of arm pulls on the *arriving arm* and the arms *stored in the memory*. The algorithm is allowed to make the following updates to the memory M:

1. Adding the arriving arm to M.
2. Discard the arriving arm, and continue to the next arriving arm.
3. Discard arm(s) from the memory M.

[4]Our upper and lower bounds apply to all sub-gaussian distributions

We define the *sample complexity* as the number of arm pulls the streaming algorithm ever uses, and the *space complexity (memory complexity)* as the maximum number of arms stored at any point (the maximum size of M). The common assumption in the literature (Assadi and Wang 2020; Maiti, Patil, and Khan 2021; Jin et al. 2021; Agarwal, Khanna, and Patil 2022; Assadi and Wang 2022) allows the algorithm to write an arbitrary number of statistics for free, i.e., do not charge costs for the size of π and any other stored information.

3 Lower Bound: A Sharp Memory-pass Trade-off for Algorithms with Known $\Delta_{[2]}$

We now introduce the construction and analysis of our main lower bound. In our lower bound proof, we will crucially use a recent multi-pass lower bound tool developed by (Assadi and Wang 2024). The original lower bound construction of (Assadi and Wang 2024) is on *batched* instances distributions. On a high level, these distributions divide the arm into multiple batches with a fixed order. Inside each batch, most of the arms are "flat", i.e., with mean reward $\frac{1}{2}$, and one (or a few) *special* arm(s) are planted uniformly at random among the indices. The reward distribution of the special arms is chosen randomly and independently by a random vriable Θ_b: if $\Theta_b = 1$, which happens with probability $f_b(B)$, then the mean rewards of the special arms are $> \frac{1}{2}$; otherwise, if $\Theta_b = 0$, the mean rewards of the special arms are $\frac{1}{2}$. If there are S special arms in batch $\mathcal{B}_b$ and $\Theta_b = 1$, the mean rewards would be $\frac{1}{2} + \eta_b^{(1)}, \frac{1}{2} + \eta_b^{(2)}, \cdots, \frac{1}{2} + \eta_b^{(S)}$. A more formal definition and the detailed description of batched instances and the lower bound results can be found in the appendix.

Our adversarial instance follows a special structure of batched instances. On a high level, our instances keep *two* special arms in each batch b with stochastic mean rewards of either $\left(\frac{1}{2}, \frac{1}{2}\right)$ or $\left(\frac{1}{2} + \eta_b^{(1)}, \frac{1}{2} + \eta_b^{(2)}\right)$. In the latter case, which happens with probability roughly $O(1/B)$, we insist on *invariate* $\eta_b^{(1)} - \eta_b^{(2)}$, which limits the utility for the knowledge of $\Delta_{[2]}$. Furthermore, we carefully pick the parameters such that the gap between $C \cdot \frac{n}{\left(\eta_b^{(1)}\right)^2}$ becomes polylog(n). Since we only work with a number of passes of $\Theta(\log(n)/\log\log(n))$, the construction allows us to "reduce" the $O\left(\sum_{i=2}^n \frac{1}{\Delta_{[i]}^2}\right)$ sample complexity to the $C \cdot \frac{n}{\left(\eta_b^{(1)}\right)^2}$ bound, which in turn allows us to use the technical lemma in (Assadi and Wang 2024) to establish the lower bound.

We now give the formal construction of the instance family.

$\mathcal{P}(B, C, \gamma)$: A hard instance distribution for multi-pass MABs algorithms with known $\Delta_{[2]}$.

1. **Parameters**: Ensure that $\frac{1}{20} \cdot \frac{1}{n^{1/3}} \leq \gamma \leq \frac{1}{10} \cdot \frac{1}{n^{1/3}}$, and let $\chi_1 = n^{1/3} \cdot \gamma$; furthermore, for any $b \in [B]$, let
$$\chi_{b+1} = \left(\frac{1}{12C\log(n)}\right)^{15} \cdot \chi_b.$$

2. **Division of arms:** Divide the n arms into $(B+1)$ batches of equal sizes, and put them in the *reverse* order of the stream, i.e. $\mathcal{B}_{B+1}$ arrives first, and $\mathcal{B}_1$ arrives the last.
3. **Sampling special arms:** For each batch $b \in [B+1]$, sample *two* arms uniformly at random (without replacement), and call them *special arms*. Set all the arms *except* the special arms with reward distribution $\text{Bern}(1/2)$.
4. **Batches $b \in [B]$:** For each $b \in [B]$, sample Θ_b from distribution $\text{Bern}(1/2B)$:
   (a) If $\Theta_b = 0$, set both special arms with reward distributions $\text{Bern}(1/2)$.
   (b) Otherwise, if $\Theta_b = 1$
   - Set the first special arm with reward distribution $\text{Bern}(1/2 + \chi_b)$.
   - Set the second special arm with reward distribution $\text{Bern}(1/2 + \chi_b + \gamma)$.
5. **The batch $B+1$:** Always set the reward distributions of the special arms as follows ($\Theta_{B+1} = 1$ deterministically)
   - Set the first special arm with reward distribution $\text{Bern}(1/2 + \chi_{B+1})$.
   - Set the second special arm with reward distribution $\text{Bern}(1/2 + \chi_{B+1} + \gamma)$.

An illustration of the distribution $\mathcal{P}(B, C, \gamma)$ can be shown in the appendix. It is straightforward to observe that the $\mathcal{P}(B, C, \gamma)$ family follows the $(B+1)$-batched instance. Furthermore, we make the crucial observation that $\Delta_{[2]}$ is invariant across different settings.

Observation 3.1. For any instance in $\mathcal{P}(B, C, \gamma)$, the value of $\Delta_{[2]}$ is equal to γ. In other words, in $\mathcal{P}(B, C, \gamma)$, for all $b \in [B+1]$, there is

$$\left(\Delta_{[2]} \mid \Theta_{<b} = 0, \Theta_b = 1\right) = \gamma.$$

We now use $\mathcal{P}(B, C, \gamma)$ to state our main multi-pass lower bound.

Theorem 1 (Formalization of Result 1)**.** *There exists a family of streaming MABs instances $\mathcal{P}$, such that any streaming algorithm (deterministic or randomized) that given the quantity of $\Delta_{[2]}$, finds the best arm from an instance sampled from $\mathcal{P}$ with an* expected *sample complexity of $O\left(\sum_{i=2}^{n} \frac{1}{\Delta_{[i]}^2} \cdot \log(n)\right)$, a success probability of at least $1999/2000$, and a memory of $o\left(n/\log^3 n\right)$ arms has to make $\Omega\left(\frac{\log(n)}{\log\log(n)}\right)$ passes over the stream.*

Limited by space, we defer the formal proof of Theorem 1 to the appendix.

4 Upper Bound: A Multi-pass Streaming MABs Algorithm with Known $\Delta_{[2]}$

A natural question to follow from our lower bound in Section 3 is whether this bound is tight. In this section, we show our main upper bound result that nearly matches our lower bound in Section 3. In particular, we prove the following theorem.

Theorem 2 (Formalization of Result 2)**.** *For any $P \geq 1$, there exists a $(P+1)$-pass streaming algorithm that given a streaming MABs instance and a known value of $\Delta_{[2]}$, finds the best arm with probability at least $1-\delta$ with a single-arm memory and at most*

$$O\left(\log\left(\frac{nP}{\delta}\right)\sum_{i=2}^{n}\frac{n^{2/P}}{\Delta_{[i]}^2}\right)$$

arm pulls.

Note that by plugging in $P = \Theta(\log(n))$, Theorem 2 gives an $O(\log(n))$-pass algorithm with $O(\sum_{i=2}^{n} \frac{1}{\Delta_{[i]}^2} \cdot \log n)$ sample complexity, as we have stated in Result 2. The psuedocodes for the algorithm is as in Algorithm 1.

Algorithm 1: The Main Multi-pass Streaming Algorithm

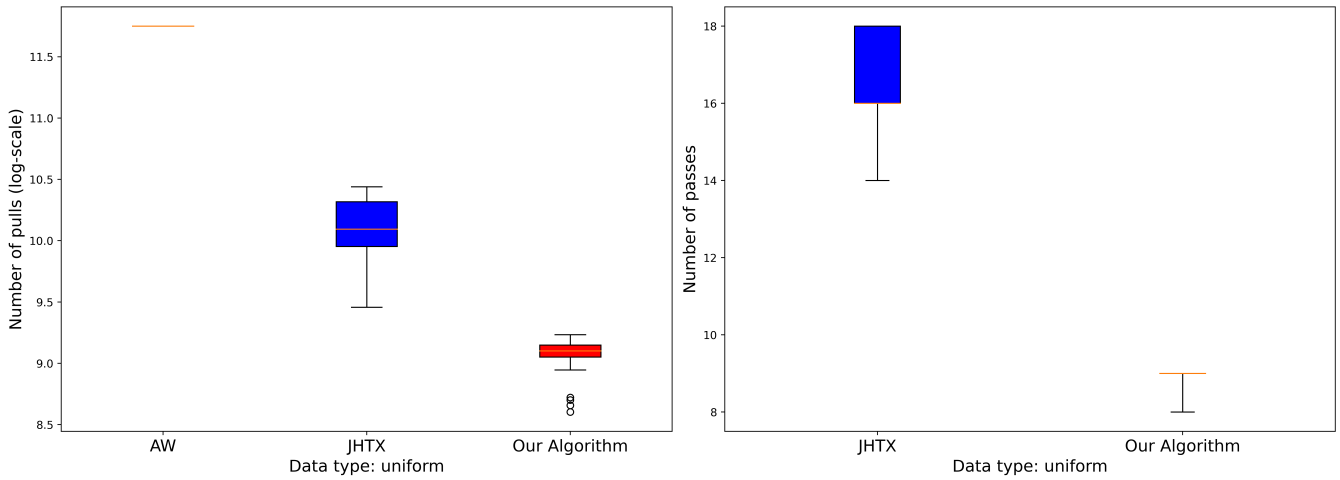
Input: Stream I, parameter P, gap parameter $\Delta_{[2]}$, and confidence parameter δ
Output: Best arm

Set $n \leftarrow |I|$ and $I_0 \leftarrow \{1, \ldots, n\}$;
Let $\epsilon_p = n^{1-p/P}\Delta_{[2]}/4$ for $p = 0, \ldots, P$;
for $p = 0, \ldots, P$ **do**
  foreach $i \in I$ *in the arrival order* **do**
    if $i \notin I_p$ **then**
      Skip arm;
    Pull arm i until the number of pulls reach $T_p \triangleq \frac{8\log(2n(P+1)/\delta)}{\epsilon_r^2 \log e}$ times;
    Compute estimated mean $\hat{\mu}_i^p$ after T_p pulls;
  Pick $\hat{\mu}_{\max}^p = \max_{i \in I_r}\{\hat{\mu}_i^p\}$;
  Create a new set $I_{p+1} \leftarrow \{i \in I_p \mid \hat{\mu}_i^p \geq \hat{\mu}_{\max}^p - \epsilon_p\}$;
if I_{P+1} *contains one element* **then**
  return *single index of arm from* I_{P+1};

Limited by space, we defer the proof of the algorithm to the appendix.

5 Experiments

We present the empirical results in this section. For multi-pass streaming MABs algorithms, there are two objectives we want to optimize: the sample complexity and the pass efficiency. Our main experimental result is that compared to existing algorithms for streaming MABs, our algorithm exhibits significant advantages on both fronts.



(a) Comparison of samples between algorithms

(b) Comparison of passes between algorithms

Figure 1: The comparison between algorithms on the sample complexity and the number of passes in the *uniform setting*. AW stands for the single-pass algorithm of (Assadi and Wang 2020), and JHTX stands for the single-pass algorithm of (Jin et al. 2021).

Experiment settings. We compare our algorithm with two benchmark algorithms: i). the AW algorithm: the single-pass algorithm by (Assadi and Wang 2020), which only uses a single pass over the stream, requires the knowledge of $\Delta_{[2]}$, and uses the worst-case optimal sample complexity of $\Theta(\frac{n}{\Delta_{[2]}^2})$; and ii). the JHTX algorithm: the $O(\log(1/\Delta_{[2]}))$-pass algorithm by (Jin et al. 2021), which does not require the knowledge of $\Delta_{[2]}$ and achieves the instance-sensitive near-instance optimal $O(\sum_{i=2}^{n} 1/\Delta_{[i]}^2 \cdot \log\log(1/\Delta_{[i]}))$ sample complexity, but uses more passes than ours. We consider instances of 2000 arms in 3 settings, and in each setting, we take 30 independent runs, and we report the error bars to avoid statistical influx. We conduct the expriments on the standard colab CPU runtime (Intel Xeon CPU with 2 vCPUs with 13GB of RAM).

Due to space limit, in this section, we only show the figures for the *uniform* setting, in which the mean rewards of the arms are from a uniform distribution supported on $[0, 1]$. Additional experiments can be found in the appendix.

Experimental results for the uniform setting. The comparison between the sample complexity and the number of passes can be found in Figure 1. Note that since the algorithm of (Assadi and Wang 2020) always uses a single pass, we do not report it in Figure 1b. It can be found that the our algorithm has the best sample complexity among the 3 algorithms. The sample complexity of the AW algorithm is considerably higher than the two others, which is understandable since in the instance with arithmetic progression means, we have $\frac{n}{\Delta_{[2]}^2} \gg \sum_{i=2}^{n} 1/\Delta_{[i]}^2 \cdot \log\log(1/\Delta_{[i]})$. comparing to the JHTX algorithm, our algorithm achieves lower mean sample complexity, and it is more stable. Finally, as we can see from Figure 1b, our algorithm uses a much smaller passes than JHTX.

6 Conclusion and Open Problems

In this paper, we established the nearly optimal bounds for multi-pass streaming MABs algorithms with a given quantity of the sub-optimality gap $\Delta_{[2]}$. We proved that to achieve the nearly instance-optimal sample complexity of $\tilde{O}(\sum_{i=2}^{n} \frac{1}{\Delta_{[2]}^2})$ with $o(n)$-arm memory, $\tilde{\Theta}(\log n)$ passes are necessary and sufficient. Our results complete a major missing piece in the pure exploration of streaming MABs, and our algorithm demonstrates strong empirical performance.

As a final remark, we note that although we worked with arms of Bernoulli distribution for both our upper and lower bounds for the convenience of presentation, our results apply to MABs with general (discrete) sub-Gaussian distributions. We can assume w.log. that the supports are on $[0, 1]$ by rescaling. For our upper bound result, we only need the Chernoff-Hoeffding inequality, which holds for all sub-Gaussian distributions. For the lower bound, since the Bernoulli distribution belongs to the sub-Gaussian family, proving lower bounds on Bernoulli arms automatically implies lower bounds for sub-Gaussian arms.

A natural open problem that follows is whether we can shave off the $\log\log(n)$ term on the number of passes on the lower bound or design an algorithm. Another interesting open problem is whether we can get a sample *lower bound* as a function of P in the same manner as our upper bound.

Acknowledgements

We thank anonymous AAAI reviewers for helpful reviews and suggestions. Part of the work was done when author Chen Wang was at Rutgers University and was supported in part by a Rutgers University Fulcrum Award.

References

Agarwal, A.; Agarwal, S.; Assadi, S.; and Khanna, S. 2017. 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*, 39–75.

Agarwal, A.; Khanna, S.; and Patil, P. 2022. A Sharp Memory-Regret Trade-off for Multi-Pass Streaming Bandits. In Loh, P.; and Raginsky, M., eds., *Conference on Learning Theory, 2-5 July 2022, London, UK*, volume 178 of *Proceedings of Machine Learning Research*, 1423–1462. PMLR.

Agarwal, D.; Chen, B.; Elango, P.; Motgi, N.; Park, S.; Ramakrishnan, R.; Roy, S.; and Zachariah, J. 2008. Online Models for Content Optimization. In *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, 17–24.

Assadi, S.; and Wang, C. 2020. Exploration with limited memory: streaming algorithms for coin tossing, noisy comparisons, and multi-armed bandits. In Makarychev, K.; Makarychev, Y.; Tulsiani, M.; Kamath, G.; and Chuzhoy, J., eds., *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, 1237–1250. ACM.

Assadi, S.; and Wang, C. 2022. Single-pass Streaming Lower Bounds for Multi-armed Bandits Exploration with Instance-sensitive Sample Complexity. In *NeurIPS*.

Assadi, S.; and Wang, C. 2024. The Best Arm Evades: Near-optimal Multi-pass Streaming Lower Bounds for Pure Exploration in Multi-armed Bandits. In *Proceedings of the 37th Conference on Learning Theory, COLT 2024*.

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

Carpentier, A.; and Locatelli, A. 2016. Tight (Lower) Bounds for the Fixed Budget Best Arm Identification Bandit Problem. In *COLT*, 590–604.

Chen, L.; and Li, J. 2015. On the Optimal Sample Complexity for Best Arm Identification. *arXiv preprint arXiv:1511.03774*.

Chen, L.; and Li, J. 2016. Open Problem: Best Arm Identification: Almost Instance-Wise Optimality and the Gap Entropy Conjecture. In Feldman, V.; Rakhlin, A.; and Shamir, O., eds., *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, volume 49 of *JMLR Workshop and Conference Proceedings*, 1643–1646. JMLR.org.

Chen, L.; Li, J.; and Qiao, M. 2017a. Nearly Instance Optimal Sample Complexity Bounds for Top-k Arm Selection. In Singh, A.; and Zhu, X. J., eds., *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*, 101–110. PMLR.

Chen, L.; Li, J.; and Qiao, M. 2017b. Towards Instance Optimal Bounds for Best Arm Identification. In Kale, S.; and Shamir, O., eds., *Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017*, volume 65 of *Proceedings of Machine Learning Research*, 535–592. PMLR.

Even-Dar, E.; Mannor, S.; and Mansour, Y. 2002. PAC Bounds for Multi-Armed Bandit and Markov Decision Processes. In *COLT*.

Hillel, E.; Karnin, Z. S.; Koren, T.; Lempel, R.; and Somekh, O. 2013. Distributed Exploration in Multi-Armed Bandits. In *NIPS*, 854–862.

Jamieson, K. G.; Malloy, M.; Nowak, R. D.; and Bubeck, S. 2014. lil' UCB : An Optimal Exploration Algorithm for Multi-Armed Bandits. In Balcan, M.; Feldman, V.; and Szepesvári, C., eds., *Proceedings of The 27th Conference on Learning Theory, COLT 2014, Barcelona, Spain, June 13-15, 2014*, volume 35 of *JMLR Workshop and Conference Proceedings*, 423–439. JMLR.org.

Jin, T.; Huang, K.; Tang, J.; and Xiao, X. 2021. Optimal Streaming Algorithms for Multi-Armed Bandits. In Meila, M.; and Zhang, T., eds., *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*, 5045–5054. PMLR.

Kalyanakrishnan, S.; and Stone, P. 2010. Efficient Selection of Multiple Bandit Arms: Theory and Practice. In *ICML*.

Karnin, Z. S.; Koren, T.; and Somekh, O. 2013. 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*, 1238–1246. JMLR.org.

Karpov, N.; Zhang, Q.; and Zhou, Y. 2020. Collaborative Top Distribution Identifications with Limited Interaction (Extended Abstract). In *FOCS*, 160–171.

Kaufmann, E.; Cappé, O.; and Garivier, A. 2016. Complexity of Best-Arm Identification in Multi-Armed Bandit Models. *Journal of Machine Learning Research*, 17(1): 1–42.

Koval, M. C.; King, J. E.; Pollard, N. S.; and Srinivasa, S. S. 2015. Robust trajectory selection for rearrangement planning as a multi-armed bandit problem. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2015, Hamburg, Germany, September 28 - October 2, 2015*, 2678–2685. IEEE.

Li, S.; Zhang, L.; Wang, J.; and Li, X. 2023. Tight Memory-Regret Lower Bounds for Streaming Bandits. *CoRR*, abs/2306.07903.

Maiti, A.; Patil, V.; and Khan, A. 2021. Multi-Armed Bandits with Bounded Arm-Memory: Near-Optimal Guarantees for Best-Arm Identification and Regret Minimization. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 19553–19565.

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

Radlinski, F.; Kleinberg, R.; and Joachims, T. 2008. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning*, 784–791.

Robbins, H. 1952. Some Aspects of the Sequential Design of Experiments. *Bulletin of the American Mathematical Society*, 58(5): 527–535.

Silva, N.; Werneck, H.; Silva, T.; Pereira, A. C.; and Rocha, L. 2022. Multi-armed bandits in recommendation systems: A survey of the state-of-the-art and future directions. *Expert Systems with Applications*, 197: 116669.

Slivkins, A. 2019. Introduction to Multi-Armed Bandits. *Found. Trends Mach. Learn.*, 12(1-2): 1–286.

Villar, S. S.; Bowden, J.; and Wason, J. 2015. 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.

Wang, C. 2023. 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.