Tight Regret Bounds for Single-pass Streaming Multi-armed Bandits

Chen Wang [1]

Abstract

Regret minimization in streaming multi-armed bandits (MABs) has been studied extensively, and recent work has shown that algorithms with $o(K)$ memory have to incur $\Omega(T^{2/3})$ regret, where K and T are the numbers of arms and trials. However, the previous best regret upper bound is still $O(K^{1/3}T^{2/3}\log^{1/3}(T))$, which is achieved by the simple uniform exploration algorithm. In this paper, we close this gap and complete the picture of regret minimization in single-pass streaming MABs. We first improve the regret lower bound to $\Omega(K^{1/3}T^{2/3})$ for algorithms with $o(K)$ memory. We then show that the $\log^{1/3}(T)$ factor is not necessary by designing algorithms with at most $O(\log^*(K))$-arm memory and achieve $O(K^{1/3}T^{2/3})$ expected regret based on streaming ε-best arm algorithms. We further tested the empirical performances of our algorithms on simulated MABs instances, where the proposed algorithms outperform the benchmark uniform exploration algorithm by a large margin and, on occasion, reduce the regret by up to 70%.

1 Introduction

The stochastic multi-armed bandits (MABs) is a classical model in machine learning and theoretical computer science that captures various real-world applications. The model was first introduced by Robbins (Robbins, 1952) for more than 70 years ago; since then, extensive research efforts have been devoted to two main problems under this model: pure exploration and regret minimization. Both problems start with a collection of K arms with unknown sub-Gaussian reward distributions. In pure exploration, we are interested in finding the best arm, defined as the arm with the highest mean reward, with as small as possible number of arm pulls (Even-Dar et al., 2002; Mannor & Tsitsiklis, 2003; Audibert et al., 2010; Karnin et al., 2013; Jamieson et al., 2014; Chen & Li, 2015; Kaufmann et al., 2016; Agarwal et al., 2017; Chen et al., 2017). On the other hand, in regret minimization, we are additionally given a parameter T as the total number of trials – also known as the 'horizon' – and we are interested in generating a plan for T arm pulls to minimize the cumulative reward gap compared to the perfect plan that puts all T pulls on the best arm (Thompson, 1933; Berry & Fristedt, 1985; Bubeck & Cesa-Bianchi, 2012; Komiyama et al., 2015; Liau et al., 2018; Slivkins, 2019; Dong et al., 2019; Chaudhuri & Kalyanakrishnan, 2020; Maiti et al., 2021; Agarwal et al., 2022). Although the two lines of research are developed relatively independently, they both have found rich applications like experiment design (Robbins, 1952; Chow & Chang, 2008), search ranking (Agarwal et al., 2008; Radlinski et al., 2008), economics (Sauré & Zeevi, 2013; Kremer et al., 2013), to name a few.

In recent years, with the strong demand to process massive data, the study of multi-armed bandits under the *streaming* model has attracted considerable attention (Liau et al., 2018; Chaudhuri & Kalyanakrishnan, 2020; Assadi & Wang, 2020; Jin et al., 2021; Maiti et al., 2021; Agarwal et al., 2022). Under this model, the arms arrive one after another in a stream, and the algorithm is only allowed to store a number of arms substantially smaller than K. In the single-pass streaming setting, if an arm is not stored or discarded from the memory, it cannot be retrieved later and is lost forever. We shall assume the order of the stream is generated by an adversary, i.e. the worst-case order. The model is a natural adaptation of the MABs to the streaming problems studied extensively in algorithms (e.g. (Alon et al., 1996; Henzinger et al., 1998; Guha et al., 2000; McGregor, 2014)).

The limited memory poses unique challenges for algorithms under this setting. Indeed, for the regret minimization application, under the classical (RAM) setting, a worst-case regret of $\Theta(\sqrt{KT})$ is necessary and achievable. However, in the single-pass streaming setting, if an algorithm is only given $o(K)$ arm memory, the recent work of (Maiti et al., 2021; Agarwal et al., 2022) proved that an $\Omega(T^{2/3})$[1] re-

[1]Department of Computer Science, Rutgers University, USA. Correspondence to: Chen Wang <chen.wang.cs@rutgers.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

[1](Maiti et al., 2021) includes another regret lower bound of $\Omega(K^{1/3}T^{2/3}/m^{7/3})$, where m is the memory of the streaming algorithm. However, their bound is only almost-tight when $m = n^{o(1)}$.

gret is inevitable. Since T could be (and is usually) much larger than K, these results already separated the regret minimization under the classical vs. the streaming settings.

Despite the progress on the lower bounds, to the best of our knowledge, there is only limited exploration on the algorithms for single-pass regret minimization. (Maiti et al., 2021) noted that a streaming implementation of the folklore uniform exploration algorithm, which pulls each arm $O((T/K)^{2/3}\log^{1/3}(T))$ times and commits to the arm with the best average empirical reward, achieves $O(K^{1/3}T^{2/3}\log^{1/3}(T))$ expected regret. Moreover, (Agarwal et al., 2022) proposed a (multi-pass) algorithm with $O(T^{2/3}\sqrt{K\log(T)})$ regret in a single pass, but it is clearly sub-optimal in the single-pass setting. As such, there remains an $O((K\log(T))^{1/3})$ gap between the upper and lower bounds.

Our Contributions. We close this gap and complete and picture for regret minimization in single-pass MABs in this work. In particular, we first tighten the regret lower bound for any algorithm with $o(K)$ memory to $\Omega(K^{1/3}T^{2/3})$ – this effectively reduces the gap between the upper and lower bounds to $\log^{1/3}(T)$. We then find that this logarithmic factor is not essential: by using an ε-best arm algorithm with $O(K/\varepsilon^2)$ arm pulls, setting $\varepsilon = (K/T)^{1/3}$, and committing to the returned arm for the rest of the trials, we can already get a regret of $O(K^{1/3}T^{2/3})$ with high constant probability.

To explore algorithms that achieve *expected* optimal regret of $O(K^{1/3}T^{2/3})$, we investigate the case when the ε-best arm algorithm *fail*. To elaborate further, the ε-arm algorithms usually succeed with *constant* probability, and by the tightness of concentration bounds, it is necessary to pay an extra $O(\log(K))$ factor if we want $1-\text{poly}(1/K)$ success probability. However, doing so will inevitably introduce an $O(\log(K))$ multiplicative factor on the regret. As such, we proceed differently by observing a smooth failure probability property for a large family of ε-best arm algorithms. On the high level, for many algorithms, even if it does not return an ε-best arm, it could still capture a 2ε-best arm with a high probability, as opposed to return an absolutely low-reward arm. We use this observation to prove a smooth-failure bounded-regret lemma, and use it to devise an algorithm with $O(K^{1/3}T^{2/3})$ expected regrets and a memory of $O(\log^*(K))$ arms.

Our results imply that the "right way" to minimize regret in single-pass streaming is to use optimal pure exploration algorithms. This establishes a connection between pure exploration and regret minimization tasks. Previous work like (Degenne et al., 2019) studied such a connection for the MABs in the offline setting; however, to the best of our knowledge, our results are the first to find the connection in the streaming setting, and it can be of independent interests.

Experiments. We evaluate the performances of the ε-best arm-based algorithm with simulated Bernoulli arms. We find that under various settings, the ε-best arm-based algorithms can consistently produce smaller regret comparing to the benchmark. In particular, we find the most stable and competitive algorithm can produce up to 70% of regret reduction, and the average regret, even account of the outliers, is at most 70% of the benchmark regret (i.e. 30% reduction). The codes of the experiment are available on github page streaming-regret-minimization-MABs.

Additional discussions about the streaming MABs model. The original motivation for (Assadi & Wang, 2020) to introduce the model was to capture the large-scale applications of MABs. For example, in the online search ranking, each arm can be viewed as a product that arrives every hour. For memory efficiency, we only want to store a limited number of products to find the best seller. We further remark that some other problems inherently require storing few arms, even when memory is not a major concern, e.g., in crowdsourcing, each arm can be viewed as a solution or a model, and storing all of them may cause management issues.

1.1 Related Work

We focus on regret minimization in streaming MABs in this work; nonetheless, it is worth mentioning that the pure exploration problem in the streaming setting also enjoys rich literature. The streaming pure exploration MABs was first introduced and studied by (Assadi & Wang, 2020), and together with the work of (Maiti et al., 2021), there are known algorithms that finds an ε-best arms with $O(\log(K))$, $O(\log\log(K))$, $O(\log^*(K))$, and $O(1)$ memory and $O(K/\varepsilon^2)$ arm pulls[2]. (Jin et al., 2021) later introduced an algorithm with a single-arm memory to find an ε-best arms with $O(K/\varepsilon^2)$ pulls, and they also studied algorithms in the multi-pass settings. The single pass pure exploration lower bound was developed recently by (Assadi & Wang, 2022). We remark that our algorithms and lower bounds are heavily inspired by the techniques developed by the aforementioned work. Furthermore, since we used ε-best arm algorithms as a subroutine in our upper bounds, our work also establish an interesting connection between the pure exploration and the regret minimization objectives.

In addition to the single-pass setting, the regret minimization problem is studied through the lens of multi-pass streams. In fact, earlier algorithms of (Liau et al., 2018; Chaudhuri & Kalyanakrishnan, 2020) all focus on regret minimization under the multi-pass settings ((Chaudhuri & Kalyanakrishnan, 2020) additionally requires random-order stream). In light of this, (Agarwal et al., 2022) provides the upper and

[2]Their $O(1)$-memory algorithm includes an additive $\Theta(\log^2(K)/\varepsilon^3)$ term.

lower regret bounds that are tight in T: they show that any P-pass algorithm with memory $o(K/P^2)$ has to incur $\Omega(T^{2^P/(2^{P+1}-1)}/2^P)$ regret; and there exists an algorithm with $O(T^{2^P/(2^{P+1}-1)}\sqrt{KP\log(T)})$ regret and $O(1)$-arm memory. Compared to their bounds, our results only apply to the single-pass, but it is tight in all asymptotic terms.

Finally, the MABs algorithms with limited memory is also explored under other models, and there are problems in the streaming setting that are closely related to MABs. For instance, (Tao et al., 2019; Karpov et al., 2020) studies the pure exploration MABs in the distributed settings, which is related to the collaborative learning with limited rounds. Furthermore, a recent line of work (Srinivas et al., 2022; Peng & Zhang, 2023) studies the streaming expert problem, where the arriving elements are the predictions from the experts. Both their model and ours have applications on online learning, yet we emphasize on different aspects.

1.2 Preliminaries

We introduce the model, the parameters, and the problem we studied in this paper in this section.

Streaming multi-armed bandits model. To begin with, we define the streaming MABs model as follows. We consider a collection of K arms with unknown sub-Gaussian reward distributions, and they arrive one after another in a stream. The algorithm can pull an arriving arm arbitrarily many times and decide whether to store it. Furthermore, the algorithm can pull a stored past arm at any point and discard some arms to free up memory when necessary. However, in the single-pass setting, an arm that is not stored or discarded is lost forever. For each arm arm_i, we let μ_i be the mean of its reward distribution. We say $\mu^* = \max_{i\in[K]} \mu_i$ is the *optimal reward* and the arm whose reward is μ^* is the *best arm*, denoted as arm^*.

Regret minimization. The regret minimization problem in stochastic multi-armed bandits goes as follows: For the regret minimization problem, we are given a fixed number of trials T (known as the *horizon*) and we want to spend as many trials as possible on the best arm. In particular, suppose algorithm $\mathcal{A}$ pulls $\mathsf{arm}_{\mathcal{A}(t)}$ in the t-th exploration, we define the *regret* of this trial as

$$r_t := \mu^* - \mu_{\mathcal{A}(t)}.$$

And we define the *total expected regret* as

$$\mathbb{E}[R_T] := \mathbb{E}\left[\sum_{t=1}^{T} \mu^* - \mu_{\mathcal{A}(t)}\right],$$

where the expectation is taken over the randomness of the arm pulls and (possibly) the algorithm. Our objective is to minimize the total expected regret. We can analogously define the minimization of probabilistic regret R_T over the randomness of the arm pulls and (possibly) the algorithm.

ε-best arm. We do *not* study ε-best arm algorithms in this paper, but rather use them as blackbox subroutines for the regret minimization purpose. In particular, an ε-best arm algorithm (also known as a $\mathsf{PAC}(\varepsilon,\delta)$ algorithm) aims to return an arm whose reward is close to μ^*. More formally, the guarantee of an ε-best arm algorithm is to output with probability at least $1-\delta$ an arm with reward μ_ε, such that $\mu^* - \mu_\varepsilon \le \varepsilon$.

Assumption of $T \ge K$. We assume w.log. in this paper that $T \ge K$, and repeatedly use this property in the proofs. Note that if $T < K$, we can easily get an upper bound of $O(T)$ by pulling an arbitrary arm for T times, and the bound is tight since with $\Omega(1)$ probability the best arm is never pulled. As such, the tight regret bound becomes $\Theta(T)$ and it is not interesting in neither theory nor practice.

Due to space limit, we defer the technical preliminaries to Appendix A.

2 The Tight Regret Lower Bound

We now formally state our lower bound result as follows.

Theorem 1. *There exists a family of streaming stochastic multi-armed bandit instances, such that for any given parameter T and K such that $T \ge K$, any single-pass streaming algorithm with a memory of $\frac{K}{20}$ arms has to suffer*

$$\mathbb{E}[R_T] \ge C \cdot K^{\frac{1}{3}} \cdot T^{\frac{2}{3}}$$

total expected regret for some constant C. Furthermore, the lower bound holds even the order of arrival for the arms is uniformly at random.

To prove Theorem 1, we will use a recent result in (Assadi & Wang, 2022) which captures a sample-space trade-off to 'trap' the best arm with limited memory.

Proposition 2.1 ((Assadi & Wang, 2022)). *Consider the following distribution of K' arms.*

> $\texttt{DIST}(K',\beta)$: ***A hard distribution with K' arms for trapping the best arm***
>
> 1. *An index $i^\star$ sampled uniform at random from $[K']$.*
> 2. *For $i \neq i^\star$, let the arms be with reward $\mu_i = \frac{1}{2}$.*
> 3. *For $i = i^\star$, let the arm be with reward $\mu_{i^\star} = \frac{1}{2} + \beta$.*

Then, any algorithm that outputs (the indices of) $\frac{K'}{8}$ arms which contains the best arm on $\texttt{DIST}$ with probability at least $\frac{2}{3}$ has to use at least $\frac{1}{1200} \cdot \frac{K'}{\beta^2}$ arm pulls.

One can refer to (Assadi & Wang, 2022) for the proof of Proposition 2.1. We note that a similar sample-space trade-off result was proved and used by (Agarwal et al., 2022) in the multi-pass setting. However, their result does not factor in the dependency on K, which creates the gap between the upper and the lower regret bounds.

Proof of Theorem 1 By Yao's minimax principle, to prove lower bounds for randomized algorithm, it suffices to consider deterministic algorithms over a certain distribution of inputs. As such, in what follows, we only consider lower bounds for deterministic algorithms over input family of instances. Our hard distribution of instances is constructed as follows.

A hard distribution for single-pass streaming MABs regret minimization

1. For the first $\frac{K}{2}$ arms, sample a set of arms from $\texttt{DIST}(K/2, \Delta)$, where $\Delta = \frac{1}{8} \cdot (\frac{K}{T})^{1/3}$.
2. For the last $\frac{K}{2}$ arms, set all arms except the last (K-th) with reward $\frac{1}{2}$.
3. The last arm follows the distribution
   (a) With probability $\frac{1}{2}$, set $\mu_K = \frac{1}{2}$;
   (b) With probability $\frac{1}{2}$, set $\mu_K = \frac{3}{4}$.

For any algorithm $\mathcal{A}$ with memory at most $\frac{K}{20}$, we analyze the two cases based on whether the algorithm uses at least $\frac{1}{2400} \cdot \frac{K}{\Delta^2}$ arm pulls on the *first* half of the stream. Note that this is the necessary number of arm pulls for $\mathcal{A}$ to store the best arm among the first half with probability at least $\frac{2}{3}$, i.e. if the algorithm uses less than the above quantity, it cannot keep the arm with reward $\frac{1}{2} + \Delta$ after the first half of the arms with probability at least $\frac{1}{3}$.

Case A). $\mathcal{A}$ uses at least $\frac{1}{2400} \cdot \frac{K}{\Delta^2}$ arm pulls on the first $\frac{K}{2}$ arms. In this case, with probability $\frac{1}{2}$, the last arm is with reward $\frac{3}{4}$. As such, each arm pull spent on the first $\frac{K}{2}$ arms incurs a regret of at least $(\frac{1}{4} - \Delta)$. As such, the expected regret is at least

$$\begin{aligned}
\mathbb{E}[R_T] &\geq \Pr(\mu_K = \frac{3}{4}) \cdot \mathbb{E}\left[R_T \mid \mu_K = \frac{3}{4}\right] \\
&\geq \frac{1}{2} \cdot \frac{1}{2400} \cdot \frac{K}{\Delta^2} \cdot (\frac{1}{4} - \Delta) \\
&\geq \frac{1}{2} \cdot \frac{1}{2400} \cdot \frac{K}{\Delta^2} \cdot \frac{1}{8} \quad (K \leq T \text{ implies } \Delta \leq \tfrac{1}{8}) \\
&= \Omega(1) \cdot K^{1/3}T^{2/3}.
\end{aligned}$$

Case B). $\mathcal{A}$ uses less than $\frac{1}{2400} \cdot \frac{K}{\Delta^2}$ arm pulls on the first $\frac{K}{2}$ arms. In this case, with probability $\frac{1}{2}$ $\texttt{arm}_K$ is with reward $\mu_K = \frac{1}{2}$; and since the memory of $\mathcal{A}$ is $K/20 < \frac{K/2}{8}$, by Proposition 2.1, with probability at least $\frac{1}{3}$, $\mathcal{A}$ does not keep the arm with reward $\frac{1}{2} + \Delta$ in the memory upon reading the $(\frac{K}{2} + 1)$-th arm. As such, we define the event

$$\mathcal{E}\text{: } \mu_K = \tfrac{1}{2} \text{ and } \mathcal{A} \text{ does } \textit{not} \text{ keep the arm with reward } \tfrac{1}{2} + \Delta \text{ after reading the first } K/2 \text{ arms}$$

and we have $\Pr(\mathcal{E}) \geq \frac{1}{6}$. Conditioning on $\mathcal{E}$, every arm pull after reading the $(\frac{K}{2} + 1)$-th arm incurs a regret of Δ, and there are at least $(T - \frac{1}{2400} \cdot \frac{K}{\Delta^2})$ trials left. As such, the expected regret is at least

$$\begin{aligned}
\mathbb{E}[R_T] &\geq \Pr(\mathcal{E}) \cdot \mathbb{E}[R_T \mid \mathcal{E}] \\
&\geq \frac{1}{6} \cdot (T - \frac{1}{2400} \cdot \frac{K}{\Delta^2}) \cdot \Delta \\
&= \frac{1}{6} \cdot (\frac{1}{8} \cdot K^{1/3}T^{2/3} - \frac{8}{2400} \cdot K^{2/3}T^{1/3}) \\
&\qquad \text{(by the choice of } \Delta) \\
&\geq \frac{1}{60} \cdot K^{1/3}T^{2/3}. \quad (K^{1/3}T^{2/3} \geq K^{2/3}T^{1/3})
\end{aligned}$$

Wrapping up the proof. Any deterministic algorithm $\mathcal{A}$ with a memory at most $\frac{K}{20}$ has to either fall in case A) or B). As such, the total expected regret is at least $C \cdot K^{1/3}T^{2/3}$ for a fixed constant C for the adversarial arrival case.

Finally, for the random order of arrival, note that by applying a random permutation to the hard distribution, with probability $\frac{1}{4}$, the arm with $\frac{1}{2} + \Delta$ is among the first $\frac{K}{2}$ arms and the arm with reward μ_K is among the latter $\frac{K}{2}$ arms. As such, by conditioning on such an event, the total expected regret becomes asymptotically the same (smaller by a $\frac{1}{4}$ factor).

3 The Tight Probabilistic Regret Upper Bound

We now turn to the upper bound results. As a first step. we show the easier result for probabilistic regret minimization: to attain the $O(K^{1/3}T^{2/3})$ regret, we only need to find an ε-best arm with $\varepsilon = (\frac{K}{T})^{1/3}$. As such, the problem can be solved in a single pass with a single-arm memory.

Theorem 2. *There exists a single-pass streaming algorithm that given a stream of stochastic multi-armed bandits and the parameters T and K, pulls the arms T times using a single-arm memory, and achieves regret*

$$R_T \leq (2\log(1/\delta) + 1) \cdot K^{\frac{1}{3}} \cdot T^{\frac{2}{3}}$$

with probability at least $1 - \delta$ over the randomness of arm pulls.

Our algorithm for Theorem 2 crucially relies on the ε-best arm algorithm in (Assadi & Wang, 2020; Jin et al., 2021).

Limited by space, we defer the detailed description and the poof of Theorem 2 to Appendix B.

Remark 1. Note that the upper bound is tight for the probabilistic regret minimization problem – the lower result in the separate note shows that for any instance in the adversarial family, the regret is at least $\Omega(K^{1/3}T^{2/3})$ with probability $\Omega(1)$. As such, we should not expect any algorithm that is asymptotically better than the guarantee of Theorem 2.

4 The Tight Expected Regret Upper Bound

The algorithm in Theorem 2 gives the optimal upper bound for regret minimization in the probabilistic manner. However, one can easily spot that the regret is not optimal in *expectation*. In fact, since the algorithms in (Assadi & Wang, 2020; Jin et al., 2021) does not provide any guarantee if the algorithm fails, if the failure probability is a constant ($\delta = \Omega(1)$), the expected regret becomes at least $\Omega(T)$. One can balance the parameters between the success and failure case to achieve an expected regret of $O(K^{1/3}T^{2/3}\log(\frac{T}{K}))$[3] – although already an improvement, the bound is still far from being tight especially when $T >> K$. As such, we need a separate investigation of the optimal algorithm for *expected regret*.

We observe that the only drawback of the exploration-and-committing strategy in Section 3 is the failure case since no guarantees is provided by existing algorithms. However, if the algorithm always keep the arm with the best empirical reward, it should not be the case that whenever the algorithm fails, it returns an absolute garbage. As such, the hope here is to obtain *smooth* probabilistic guarantees from existing ε-best arm algorithms to attain the optimal regret bound.

In what follows, we proceed our main upper bound result by first showing that if the smooth probabilistic guarantee holds, we can indeed obtain algorithms with low regret (Lemma 4.1). Subsequently, we present two algorithm with expected regret $O(K^{1/3}T^{2/3}\log(K))$ and $O(K^{1/3}T^{2/3})$, respectively. Both bounds utilize ε-best arm algorithms as subroutines – the first bound employs a variate of the simple naive uniform elimination algorithm, while the second bound uses a more involved algorithm by (Assadi & Wang, 2020) and (Maiti et al., 2021).

4.1 A Smooth-Failure Bounded-Regret Lemma

We first present a technical lemma that gives a regret upper bound provided an ε-best arm algorithms that display a 'smooth trade-off' between the arm reward and the failure probability. The formal statement of the lemma is as follows.

Lemma 4.1 (Smooth-Failure Bounded-Regret Lemma)**.** *Let* INST *be a streaming multi-armed bandit instance with fixed parameters T, K such that $T > K$, and let* ALG *be a streaming algorithm that given parameter ε, uses S space and $\frac{M}{\varepsilon^2}$ arm pulls to returns an* $\mathsf{arm}_{\mathsf{ALG}(\mathit{INST})}$ *such that*

$$\Pr\left(\mu_{\mathsf{ALG}(\mathit{INST})} < \mu^* - c\cdot\varepsilon\right) \le (\frac{1}{2})^c \cdot \frac{1}{10}.$$

for any integer $c \ge 1$. Then, there exists an S-space streaming algorithm that achieves $O(\frac{M}{K^{2/3}}T^{2/3} + K^{1/3}T^{2/3})$ regret in expectation, i.e.

$$\mathbb{E}[R_T] \le O(M\cdot\frac{T^{2/3}}{K^{2/3}} + K^{1/3}T^{2/3}).$$

Proof. The algorithm is to simply run the streaming algorithm for the ε-best arm with $\varepsilon = (\frac{K}{T})^{1/3}$ (the exploration phase), and commit to the returned arm $\mathsf{arm}_{\mathsf{ALG(INST)}}$ for the rest of the trials if there is any remaining trials (the committing phase). As such, the space bound trivially follows since we do not use any extra space.

We now analyze the expected regret. To proceed, we let R_T^e be the regret induced by the exploration phase, and R_T^c be the regret induced by the committing phase. By the choice of the parameter ε, we (deterministically) have $R_T^e \le \frac{M}{\varepsilon^2} = M\cdot\frac{T^{2/3}}{K^{2/3}}$, which implies

$$\mathbb{E}[R_T^e] \le M\cdot\frac{T^{2/3}}{K^{2/3}}.$$

Hence, we only need to control $\mathbb{E}[R_T^c]$ by the linearity of expectation. To continue, we define the events

$$\mathcal{E}_{c,\varepsilon} = \text{The algorithm finds an arm with reward at least } \mu^* - c\cdot\varepsilon$$

for every integer $c \ge 1$. Observe that an event $\mathcal{E}_{c,\varepsilon}$ *contain* all events with $\mathcal{E}_{c',\varepsilon}$ for $c' < c$. As such, using $\mathcal{E}_{1:c-1,\varepsilon}$ as a short-hand notation of the *collection* of events from $\mathcal{E}_{1,\varepsilon}$ to $\mathcal{E}_{c-1,\varepsilon}$, we note that $\neg\mathcal{E}_{c-1,\varepsilon}$ means *none* of the event from $\mathcal{E}_{1,\varepsilon}$ to $\mathcal{E}_{c-1,\varepsilon}$ happens. As such, we can re-write the expected regret as follows.

$$\begin{aligned}
&\mathbb{E}[R_T^c]\\
&= \mathbb{E}[R_T^c \mid \mathcal{E}_{1,\varepsilon}]\Pr(\mathcal{E}_{1,\varepsilon})\\
&\quad + \mathbb{E}[R_T^c \mid \neg\mathcal{E}_{1,\varepsilon}]\Pr(\neg\mathcal{E}_{1,\varepsilon})\\
&= \mathbb{E}[R_T^c \mid \mathcal{E}_{1,\varepsilon}]\Pr(\mathcal{E}_{1,\varepsilon})\\
&\quad + \Pr(\neg\mathcal{E}_{1,\varepsilon})\cdot\mathbb{E}[R_T^c \mid \mathcal{E}_{2,\varepsilon},\neg\mathcal{E}_{1,\varepsilon}]\Pr(\mathcal{E}_{2,\varepsilon}\mid\neg\mathcal{E}_{1,\varepsilon})\\
&\quad + \Pr(\neg\mathcal{E}_{1,\varepsilon})\cdot\mathbb{E}[R_T^c \mid \neg\mathcal{E}_{2,\varepsilon}]\Pr(\neg\mathcal{E}_{2,\varepsilon}\mid\neg\mathcal{E}_{1,\varepsilon})\\
&= \cdots\\
&= \mathbb{E}[R_T^c \mid \mathcal{E}_{1,\varepsilon}]\Pr(\mathcal{E}_{1,\varepsilon})\\
&\quad + \sum_{c=2}^{\infty}\mathbb{E}[R_T^c \mid \mathcal{E}_{c,\varepsilon},\neg\mathcal{E}_{c-1,\varepsilon}]\Pr(\mathcal{E}_{c,\varepsilon},\neg\mathcal{E}_{c-1,\varepsilon}).
\end{aligned}$$

[3] Concretely, by setting $\delta = (\frac{K}{T})^{1/3}$, the expected regret is $O\left((1-(\frac{K}{T})^{1/3})\cdot\log\left((\frac{K}{T})^{1/3}\right)K^{1/3}T^{2/3} + T\cdot(\frac{K}{T})^{1/3}\right)$, which is upper-bounded by $O(K^{1/3}T^{2/3}\log(\frac{T}{K}))$.

Since $\varepsilon = (\frac{K}{T})^{1/3}$, conditioning on event $\mathcal{E}_{c,\varepsilon}$ happens, the regret induced by the committing part is at most

$$\begin{aligned} R_T^c \mid \mathcal{E}_{c,\varepsilon}, \neg\mathcal{E}_{c-1,\varepsilon} &= c \cdot (\frac{K}{T})^{1/3} \cdot T \\ &\leq c \cdot K^{1/3}T^{2/3}. \end{aligned}$$

On the other hand, recall that the probability for each $\neg\mathcal{E}_{c,\varepsilon}$ is at most $\frac{1}{2^c} \cdot \frac{1}{10}$. As such, the probability for $\Pr(\mathcal{E}_{c,\varepsilon}, \neg\mathcal{E}_{c-1,\varepsilon})$ can be bounded as

$$\begin{aligned} \Pr(\mathcal{E}_{c,\varepsilon}, \neg\mathcal{E}_{c-1,\varepsilon}) &= \Pr(\mathcal{E}_{c,\varepsilon} \mid \neg\mathcal{E}_{c-1,\varepsilon}) \cdot \Pr(\neg\mathcal{E}_{c-1,\varepsilon}) \\ &\leq \Pr(\neg\mathcal{E}_{c-1,\varepsilon}) \qquad (\Pr(\mathcal{E}_{c,\varepsilon} \mid \neg\mathcal{E}_{c-1,\varepsilon}) \leq 1) \\ &\leq (\frac{1}{2})^{c-1} \cdot \frac{1}{10}. \end{aligned}$$

As such, the expected regret of the committing phase can be bounded as a convergent summation of terms:

$$\begin{aligned} \mathbb{E}[R_T^c] &\leq \frac{1}{10} \cdot K^{1/3}T^{2/3} \cdot \sum_{c=1}^{\infty} \frac{c}{2^{c-1}} \\ &= \frac{2}{5} \cdot K^{1/3}T^{2/3}. \qquad (\textstyle\sum_{c=1}^{\infty} \frac{c}{2^{c-1}} = 4) \end{aligned}$$

Therefore, we have the expected regret to be

$$\begin{aligned} \mathbb{E}[R_T] &= \mathbb{E}[R_T^e + R_T^c] \qquad \text{(linearity of expectation)} \\ &\leq M \cdot \frac{T^{2/3}}{K^{2/3}} + \frac{2}{5} \cdot K^{1/3}T^{2/3} \\ &= O\left(M \cdot \frac{T^{2/3}}{K^{2/3}} + K^{1/3}T^{2/3}\right), \end{aligned}$$

as desired. □

Lemma 4.1 provides a neat approach to bound the expected regret by bounding the number of arm pulls and 'smooth failure probability' for ε-best arm algorithms. As we will see shortly, algorithms based on Chernoff bound generally satisfy the smooth failure probability. Note that, however, streaming algorithms based on amortized variance analysis (e.g. the single-arm algorithm in (Assadi & Wang, 2020)) do *not* generally satisfy this property.

4.2 A $\log^*(K)$-arm memory algorithm with $O(K^{1/3}T^{2/3})$ expected regret

If we run the naive elimination of ε-best arm, we can get an algorithm with $O(K^{1/3}T^{2/3}\log(K))$ regret with the memory of a single arm. Limited by space, we defer the description of this algorithm to Appendix C. For now, we proceed to our streaming algorithm with the *optimal* expected regret for any streaming algorithm with $O(\log^*(K))$ memory. Our optimal algorithm follows the same 'exploration-and-commit' paradigm, albeit using a non-trivial streaming ε-best arm algorithm recently developed by (Assadi & Wang, 2020; Maiti et al., 2021).

We first give the streaming ε-best arm algorithm with $\log^*(K)$ memory as follows.

Parameter Set 1:

$\{\varepsilon\}_{\ell\geq 1} : \varepsilon_\ell = \frac{\varepsilon}{10 \cdot 2^{\ell-1}}$ (ε parameter at each level)

$\{r_\ell\}_{\ell\geq 1} : r_1 := 4, \quad r_{\ell+1} = 2^{r_\ell}$;

$\{\beta_\ell\}_{\ell\geq 1} : \beta_\ell = \frac{1}{\varepsilon_\ell^2}$;

(intermediate variables to define s_ℓ and c_ℓ)

$\{s_\ell\}_{\ell\geq 1} : s_\ell = 8\beta_\ell(\ln(\frac{1}{\delta}) + 3r_\ell)$

(number of samples per arm at each level)

$\{c_\ell\}_{\ell\geq 1}\, c_1 = 2^{r_1}, \quad c_\ell = \frac{2^{r_\ell}}{2^{\ell-1}} (\ell \geq 2)$

(the bound on number of arms to 'defeat' at each level)

Aggressive Selective Promotion – an ε-best arm algorithm using $\log^*(K)$-arm memory

Counters: $C_1, C_2, ..., C_t \qquad t = \lceil \log^*(K) \rceil + 1$;
Reward records: $\mu_1^*, \mu_2^*, ..., \mu_t^*$, initialize with 0;
Stored arms: $\mathsf{arm}_1^*, \mathsf{arm}_2^*, ..., \mathsf{arm}_t^*$ the most reward arm of ℓ-th level.

- For each arriving arm_i in the stream do:
  - (1) Read arm_i to memory.
  - (2) Starting from level $\ell = 1$:
    - (a) Sample arm_i for s_ℓ times and get $\widehat{\mu}_{\mathsf{arm}_i}$.
      - i. If $\widehat{\mu}_{\mathsf{arm}_i} < \mu_\ell^*$, drop arm_i;
      - ii. Otherwise, replace arm_ℓ^* with arm_i and set $\mu_\ell^* = \widehat{\mu}_{\mathsf{arm}_i}$.
    - (b) Increase C_ℓ by 1.
    - (c) If $C_\ell = c_\ell$, do
      - i. Reset the counter to $C_\ell = 0$.
      - ii. Send arm_ℓ^* to the next level by calling Line 2((3))i with $(\ell = \ell + 1)$.
  - (3) At the end of the stream
    - (a) For all $i \in [t]$, sample arm_i^* for $32 \cdot \frac{\log^*(K)}{\varepsilon^2}$ times and get $\widehat{\mu}_i^*$.
    - (b) Return the arm with the highest $\widehat{\mu}_i^*$.

Unlike the Naive Uniform Elimination algorithm, it is not immediately clear how many arm pulls are used in the Aggressive Selective Promotion algorithm. We can nevertheless use the upper bound on arm pulls in (Assadi & Wang, 2020; Maiti et al., 2021) as a blackbox.

Lemma 4.2 ((Assadi & Wang, 2020; Maiti et al., 2021)). *The number of arm pulls used by the Aggressive Selective Promotion algorithm is* $O(\frac{K}{\varepsilon^2}\log(\frac{1}{\delta}))$.

Note that Lemma 4.2 holds deterministically without any randomness – this is simply because of the number of arms reaching higher levels decreases in a towering number speed. On the other hand, it is not immediately clear which arm the Aggressive Selective Promotion algorithm will return if it fails. To this end, we again prove a 'smooth version' of success probability for the Aggressive Selective Promotion algorithm.

Lemma 4.3. *For fixed parameters* $\delta \in (0,1)$, $\varepsilon \in (0,1)$, *and integer* $c \geq 1$, *the Aggressive Selective Promotion algorithm returns an* arm^*_t *with reward*

$$\mu_{\mathsf{arm}^*_t} \geq \mu^* - c\cdot\varepsilon$$

with probability at least $1-(\frac{1}{2})^{c^2}\cdot\delta$.

Proof. Fix a level ℓ, we define the surviving arms of level ℓ as the set of arms that can ever reach ℓ, and let the corresponding mean reward be μ_ℓ (pending the randomness of the arms). Our strategy is to argue that with probability at least $\left(1-(\frac{1}{2})^{c^2+2\ell}\cdot\delta\right)$, the best arm among the surviving arms of level ℓ can only be replaced by an arm with mean reward at least $\mu_\ell - c\cdot\varepsilon_\ell$. Since arm^* is trivially the best arm among the surviving arms of level 1, this allows us to guarantee the cumulative gap as a summation of $c\dot{\varepsilon}_\ell$ across levels – a series that converges $c\cdot\varepsilon$.

We now formalize the above strategy. We first show at any level ℓ, the value of the 'benchmark' μ^*_ℓ does not go below $\mu_\ell - \frac{c}{2}\cdot\varepsilon_\ell$ with probability at least $\left(1-(\frac{1}{2})^{c^2+3r_\ell}\cdot\delta\right)$. To see this, note that by an application of Lemma A.2, for any arm with mean reward μ, there is

$$\begin{aligned}\Pr\left(\widehat{\mu}\leq \mu - c\cdot\varepsilon_\ell/2\right) &\leq \exp\left(-2c^2\cdot(\log(\frac{1}{\delta})+3r_\ell)\right)\\ &\qquad\text{(arm is pulled } s_\ell = 8\beta_\ell(\ln(\tfrac{1}{\delta})+3r_\ell)\text{ times)}\\ &\leq (\frac{1}{2})^{c^2+3r_\ell}\cdot\delta.\end{aligned}$$

As such, let μ_ℓ be the mean reward of the best surviving arm of level ℓ, the empirical reward for μ_ℓ is at least $\mu_\ell - \frac{c}{2}\cdot\varepsilon_\ell$. Suppose the value of μ^*_ℓ (the benchmark reward) is less than $\mu_\ell - \frac{c}{2}\cdot\varepsilon_\ell$; then, when μ_ℓ joins level ℓ, the benchmark is updated to the value with probability at least $1-(\frac{1}{2})^{c^2+3r_\ell}\cdot\delta$.

We then show that at any level ℓ, any arm with reward less than $\mu_\ell - \varepsilon_\ell$ can have a empirical reward of at most $\mu_\ell - \frac{\varepsilon_\ell}{2}$ with probability $1-(\frac{1}{2})^{c^2+2r_\ell}\cdot\delta$, again by an application of Lemma A.2. For an arm with reward μ, there is

$$\begin{aligned}\Pr\left(\widehat{\mu}\geq \mu + c\cdot\varepsilon_\ell/2\right) &\leq \exp\left(-2c^2\cdot(\log(\frac{1}{\delta})+3r_\ell)\right)\\ &\qquad\text{(arm is pulled } s_\ell = 8\beta_\ell(\ln(\tfrac{1}{\delta})+3r_\ell)\text{ times)}\\ &\leq (\frac{1}{2})^{c^2+3r_\ell}\cdot\delta.\end{aligned}$$

As such, we can apply a union bound over the bad events, and obtain that

$$\begin{aligned}&\Pr\left(\widehat{\mu}\geq \mu + c\cdot\varepsilon_\ell/2 \text{ for any arm on level } \ell\right)\\ &\leq c_\ell\cdot(\frac{1}{2})^{c^2+3r_\ell}\cdot\delta \leq (\frac{1}{2})^{c^2+2r_\ell}\cdot\delta.\end{aligned}$$

For any integer c, we now have the following statement: by a union bound, with probability at least

$$1-\left((\frac{1}{2})^{c^2+2r_\ell}+(\frac{1}{2})^{c^2+3r_\ell}\right)\cdot\delta \geq 1-(\frac{1}{2})^{c^2+2\ell}\cdot\delta,$$

the benchmark reward on level ℓ is at least $\mu_\ell - c\cdot\frac{\varepsilon_\ell}{2}$, and an arm with such an empirical reward has to have a mean reward of at least $\mu_\ell - c\cdot\varepsilon_\ell$. Therefore, we conclude that at a fixed level ℓ and for any integer c, the best arm^*_ℓ has to have a mean reward at least $\mu_\ell - c\cdot\varepsilon_\ell$ with probability at least $1-(\frac{1}{2})^{c^2+2\ell}\cdot\delta$. We define this high-probability event at level ℓ as $\mathcal{A}_\ell$.

Finally, we handle the accumulation of error and failure probability across levels. Note that the failure probability across different levels can be bounded by

$$\begin{aligned}\Pr\left(\neg\mathcal{A}_\ell \text{ at any level } \ell\right) &\leq \sum_{\ell=1}^{t}(\frac{1}{2})^{c^2+2\ell}\cdot\delta\\ &\leq (\frac{1}{2})^{c^2}\delta\sum_{\ell=1}^{\infty}(\frac{1}{2})^{2\ell}\\ &\leq (\frac{1}{2})^{c^2}\cdot\delta.\end{aligned}$$

Conditioning on the high probability event over all levels of ℓ, the cumulative gap between the best surviving arm on level 1 (which is arm^*) and on level t is at most

$$\begin{aligned}\sum_{\ell=1}^{t}c\cdot\varepsilon_\ell &= c\cdot\sum_{\ell=1}^{\infty}\varepsilon_\ell\\ &\leq c\cdot\frac{\varepsilon}{30}\sum_{\ell=1}^{\infty}\frac{1}{2^{\ell-1}}\\ &\leq c\cdot\varepsilon,\end{aligned}$$

as desired by the lemma statement. □

We can now arrive at our main $\log^*(K)$-memory regret minimization algorithm by combining Lemmas 4.1 to 4.3.

Theorem 3. *There exists a single-pass streaming algorithm that given a multi-armed bandit instance arriving in a stream with fixed parameters T, K such that $T > K$, carries out arm pulls with expected regret $\mathbb{E}[R_T] \leq O(K^{1/3}T^{2/3})$ and uses a memory of $\lceil \log^*(K) \rceil + 1$ arms.*

Proof. By Lemma 4.3, for any given parameter ε, there is

$$\Pr\left(\mu_{\mathsf{arm}_t^*} < \mu^* - c \cdot \varepsilon\right) \leq (\frac{1}{2})^{c^2} \cdot \frac{1}{10} \leq (\frac{1}{2})^{c} \cdot \frac{1}{10}.$$

by setting $\delta = \frac{1}{10}$. As such, we can match the parameters in Lemma 4.1 by $S = \lceil \log^*(K) \rceil + 1$ and $M = O(K)$ as in Lemma 4.2. This gives us the desired bound of

$$\mathbb{E}[R_T] \leq O\left(M \cdot \frac{T^{2/3}}{K^{2/3}} + K^{1/3}T^{2/3}\right) = O(K^{1/3}T^{2/3}),$$

which is asymptotically optimal for any streaming algorithm with $o(K)$-arm memory. □

Remark 2. In (Assadi & Wang, 2020), there are additional algorithms with $\log(K)$- and $\log\log(K)$-arm memory that find ε-best arms with $O(\frac{K}{\varepsilon^2})$ arm pulls. Since they follow the same paradigm to apply concentration bounds as in *Aggressive Selective Promotion*, it can be shown that they can also be converted to regret minimization algorithms with the *optimal expected regret*. We provide their algorithmic description in Appendix E.2 without proofs since they are very similar to Lemma 4.3. We remark that although the memory bounds are worse, for practical implementation, their regret could be smaller than the Aggressive Selective Promotion, and the memory difference is not significant up to 10^{10} arms. We will see more on this in Section 5.

A discussion about the single-arm algorithm. One may naturally wonder whether we can achieve a single-arm memory by the smooth-failure bounded regret lemma – after all, we are using known algorithms, and the main innovation lies in the analysis. Alas, it appears that at least the single-arm algorithm in (Assadi & Wang, 2020) does not follow the property. At a high level, the single-arm algorithm (and a variate that stored 2 arms, both known as GAME-OF-ARMS) in (Assadi & Wang, 2020) uses the ideas of (i). a multi-level challenge with a geometrically increasing number of arm pulls, and (ii) a "budget" the number of arm pulls that is used for a stored arm. They proved that if the stored arm is sufficiently good, say it is the best arm, then with probability at least $99/100$ (or some arbitrary $1 - \delta$ by paying $\log(1/\delta)$), the number of arm pulls we used will never exceed a (varying) budget. As such, we can discard an arm whose arriving "challengers" uses a large number of arm pulls if we only want to find the best arm with high constant probability.

However, for the expected regret minimization task, with probability $\sim 1/100$, the best arm can actually be discarded, and the algorithm may return an arbitrary arm. One can think of an adversarial instance that uses a considerable number of arms with suboptimal yet "high enough" rewards that "almost exhaust" the sample bound of the stored best arm; then a very bad arm (say with reward 0.0001) comes but still manages to break the sample budget with a small constant probability. Now, the algorithm may commit to this arm, and the expected regret becomes at least $T >> K$. Therefore, it is not immediately clear whether we can achieve $O(1)$-arm for the expected regret minimization in a single pass, and it is an interesting direction to pursue.

5 Implementation and Simulation Results

In this section, we show the empirical evaluation of our algorithms under simulations on Bernoulli arms. In particular, we implemented and tested the uniform exploration algorithm, the naive uniform elimination algorithm, the algorithms from ε-best arm with $O(\log(K))$, $O(\log\log(K))$ and $O(\log^*(K))$ memory, and the 2-arm GAME-OF-ARMS algorithm as in (Assadi & Wang, 2020). The uniform exploration algorithm is used as the benchmark as it is the known best regret minimization algorithm with provable guarantees in a single pass.

Our simulation results find that the proposed algorithm in this paper outperforms the baseline *by a significant margin*. Under all of our setting (each with 50 runs), there is at least one ε-best arm-based algorithm that achieves 80% of the benchmark regret on average and 70% on median, and the margin can be as significant as 70% of the benchmark when T is large (i.e. 30% regret of the benchmark). Across different settings, the best algorithm (the $O(\log\log(K))$-space algorithm) outperforms the uniform exploration algorithm by around 30% of the mean regret (i.e. 70% of the benchmark mean regret) and $> 50\%$ of the median regret (i.e. $< 50\%$ of the benchmark mean regret), while all the ε-best arm-based algorithm outperforms the uniform exploration and the naive elimination in most cases. Interestingly, the 2-arm GAME-OF-ARMS algorithm offers competitive performances, despite being theoretically sub-optimal in (worst-case instance) expected regret.

Limited by space, we only show the experimental results for one of the settings in Table 1 and Figure 1, where the number of arms is set to $K = 50000$, and the number of arm pulls are tested with $1000K$, $1000K^2$ and $1000K^3$ [4]. The means of the reward distributions in each instance are sampled uniformly from $[0, 1]$, and we include 50 runs in each setting. The regrets in the table is of the *relative scale*: we treat the regret of the uniform exploration algorithm as 1.0 for benchmark. From the table and the figure, we

[4] In the figures, we use $\log(n)$, $\log\log(n)$, and $\log^*(n)$ (using notation of n instead of K) as type of algorithms to keep consistent with the original algorithms in the pure exploration context.

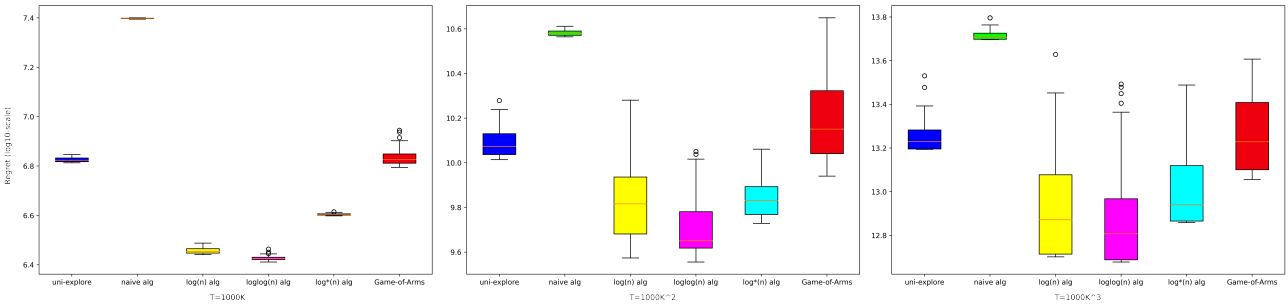

Figure 1. The regret error bars for $K = 50000$ uniform reward setting of the stream.

Table 1. The comparison of the relative regret for different algorithms under setting $K = 50000$ uniform stream setting.

	Uniform Exploration	Naive Elimination	$\log(K)$ ε-best	$\log\log(K)$ ε-best	$\log^*(K)$ ε-best	Game-of-Arms
Mean Regret						
$T = 1000K$	1.0	3.7355	0.4274	0.4000	0.6012	1.0290
$T = 1000K^2$	1.0	3.0423	0.5989	0.4374	0.5733	1.2976
$T = 1000K^3$	1.0	2.8652	0.5686	0.5117	0.5982	1.0960
Median Regret						
$T = 1000K$	1.0	3.7555	0.4264	0.3994	0.6036	1.0008
$T = 1000K^2$	1.0	3.1974	0.5525	0.3776	0.5713	1.1953
$T = 1000K^3$	1.0	3.0008	0.4393	0.3789	0.5142	0.9996

can observe the competitive performances offered by the $\log(K)$-, $\log\log(K)$-, and $\log^*(K)$-memory algorithms.

We defer the full details of the experiments and the discussions to Appendix D.

6 Conclusion

In this paper, we studied the tight lower and upper bounds for regret minimization for single-pass streaming multi-armed bandits. In particular, we first improved the regret lower bound for streaming algorithms with $o(K)$ memory from $\max\{\Omega(T^{2/3}), \Omega(K^{1/3}T^{2/3}/m^{7/3})\}$ to $\Omega(K^{1/3}T^{2/3})$, which is tight in both T and K. We then proved that the $\Theta(K^{1/3}T^{2/3})$ regret, with high (constant) probability, can be achieved by adopting an ε-best arm algorithm with $O(K/\varepsilon^2)$ arm pulls, setting the parameter $\varepsilon = (K/T)^{1/3}$, and committing to the returned arm. Finally, we showed that the simple exploration-and-commit strategy can achieve the *expected* optimal regret of $\Omega(K^{1/3}T^{2/3})$ with a large family of streaming ε-best arm algorithms, and the memory can be as small as $O(\log^*(K))$. We empirically tested the performances of the ε-best arm-based algorithms on simulations of MABs streams, and we found that the proposed algorithms can significantly outperform the benchmark uniform exploration algorithm.

Our work completes the picture for regret minimization in single-pass streaming MABs with *sublinear* arm memory. On the other hand, it also opens several directions of open problems for future exploration. The first question is whether the memory of arms can be further reduced to $O(1)$ or a single arm, as did in the pure exploration algorithms of (Assadi & Wang, 2020) and (Jin et al., 2021). Note that the single-arm memory algorithm in (Assadi & Wang, 2020) may actually return a very bad arm, and it is unclear whether the algorithm in (Jin et al., 2021) has the smooth-failure property. Another open question is the multi-pass setting, where (Agarwal et al., 2022) proved tight bounds for regrets minimization with sublinear arm memory as a function of T, but the tight dependent on K is still unclear. Finally, it will be interesting to see the application of our algorithms in real-world scenarios.

Acknowledgement

The author would like to thank Michael Saks and Sepehr Assadi of Rutgers University for very helpful discussions and anonymous ICML 2023 reviewers for their constructive comments. This research is supported in part by NSF CAREER Grant CCF-2047061, a gift from Google Research, and a Rutgers Research Fulcrum Award.

References

Agarwal, A., Agarwal, S., Assadi, S., and Khanna, S. 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*, pp. 39–75, 2017. 1

Agarwal, A., Khanna, S., and Patil, P. 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*, pp. 1423–1462. PMLR, 2022. URL https://proceedings.mlr.press/v178/agarwal22a.html. 1, 2, 4, 9

Agarwal, D., Chen, B., Elango, P., Motgi, N., Park, S., Ramakrishnan, R., Roy, S., and Zachariah, J. 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*, pp. 17–24, 2008. 1

Alon, N., Matias, Y., and Szegedy, M. The space complexity of approximating the frequency moments. In *STOC*, pp. 20–29. ACM, 1996. 1

Assadi, S. and Wang, C. 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*, pp. 1237–1250. ACM, 2020. doi: 10.1145/3357713.3384341. URL https://doi.org/10.1145/3357713.3384341. 1, 2, 4, 5, 6, 7, 8, 9, 13, 19, 22, 23

Assadi, S. and Wang, C. 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. 2, 3, 4

Audibert, J., Bubeck, S., and Munos, R. Best arm identification in multi-armed bandits. In *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*, pp. 41–53, 2010. 1

Berry, D. A. and Fristedt, B. Bandit problems: sequential allocation of experiments (monographs on statistics and applied probability). *London: Chapman and Hall*, 5 (71-87):7–7, 1985. 1

Bubeck, S. and Cesa-Bianchi, N. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Found. Trends Mach. Learn.*, 5(1):1–122, 2012. doi: 10.1561/2200000024. URL https://doi.org/10.1561/2200000024. 1

Chaudhuri, A. R. and Kalyanakrishnan, S. 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*, pp. 10085–10092, 2020. 1, 2

Chen, L. and Li, J. On the optimal sample complexity for best arm identification. *CoRR*, abs/1511.03774, 2015. URL http://arxiv.org/abs/1511.03774. 1

Chen, L., Li, J., and Qiao, M. Towards instance optimal bounds for best arm identification. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017*, pp. 535–592, 2017. 1

Chow, S.-C. and Chang, M. Adaptive design methods in clinical trials–a review. *Orphanet journal of rare diseases*, 3(1):1–13, 2008. 1

Degenne, R., Nedelec, T., Calauzènes, C., and Perchet, V. Bridging the gap between regret minimization and best arm identification, with application to A/B tests. In Chaudhuri, K. and Sugiyama, M. (eds.), *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pp. 1988–1996. PMLR, 2019. URL http://proceedings.mlr.press/v89/degenne19a.html. 2

Dong, S., Ma, T., and Roy, B. V. On the performance of thompson sampling on logistic bandits. In *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, pp. 1158–1160, 2019. 1

Even-Dar, E., Mannor, S., and Mansour, Y. PAC bounds for multi-armed bandit and markov decision processes. In *Computational Learning Theory, 15th Annual Conference on Computational Learning Theory, COLT 2002, Sydney, Australia, July 8-10, 2002, Proceedings*, pp. 255–270, 2002. 1, 14, 15

Guha, S., Mishra, N., Motwani, R., and O'Callaghan, L. Clustering data streams. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pp. 359–366, 2000. 1

Henzinger, M. R., Raghavan, P., and Rajagopalan, S. Computing on data streams. In *External Memory Algorithms,*

Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, May 20-22, 1998, pp. 107–118, 1998. 1

Jamieson, K. G., Malloy, M., Nowak, R. D., and Bubeck, S. lil' UCB : An optimal exploration algorithm for multi-armed bandits. In *Proceedings of The 27th Conference on Learning Theory, COLT 2014, Barcelona, Spain, June 13-15, 2014*, pp. 423–439, 2014. 1

Jin, T., Huang, K., Tang, J., and Xiao, X. 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*, pp. 5045–5054. PMLR, 2021. URL http://proceedings.mlr.press/v139/jin21a.html. 1, 2, 4, 5, 9, 13, 14, 22

Karnin, Z. S., Koren, T., and Somekh, O. 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*, pp. 1238–1246, 2013. 1

Karpov, N., Zhang, Q., and Zhou, Y. Collaborative top distribution identifications with limited interaction (extended abstract). In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pp. 160–171. IEEE, 2020. doi: 10.1109/FOCS46700.2020.00024. URL https://doi.org/10.1109/FOCS46700.2020.00024. 3

Kaufmann, E., Cappé, O., and Garivier, A. On the complexity of best-arm identification in multi-armed bandit models. *J. Mach. Learn. Res.*, 17:1:1–1:42, 2016. 1

Komiyama, J., Honda, J., and Nakagawa, H. Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with multiple plays. In *International Conference on Machine Learning*, pp. 1152–1161. PMLR, 2015. 1

Kremer, I., Mansour, Y., and Perry, M. Implementing the "wisdom of the crowd". In Kearns, M. J., McAfee, R. P., and Tardos, É. (eds.), *Proceedings of the fourteenth ACM Conference on Electronic Commerce, EC 2013, Philadelphia, PA, USA, June 16-20, 2013*, pp. 605–606. ACM, 2013. doi: 10.1145/2492002.2482542. URL https://doi.org/10.1145/2492002.2482542. 1

Liau, D., Song, Z., Price, E., and Yang, G. Stochastic multi-armed bandits in constant space. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, pp. 386–394, 2018. 1, 2

Maiti, A., Patil, V., and Khan, A. 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*, pp. 19553–19565, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/a2f04745390fd6897d09772b2cd1f581-Abstract.html. 1, 2, 5, 6, 7

Mannor, S. and Tsitsiklis, J. N. Lower bounds on the sample complexity of exploration in the multi-armed bandit problem. In *Computational Learning Theory and Kernel Machines, 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003, Proceedings*, pp. 418–432, 2003. 1

McGregor, A. Graph stream algorithms: a survey. *SIGMOD Rec.*, 43(1):9–20, 2014. doi: 10.1145/2627692.2627694. URL https://doi.org/10.1145/2627692.2627694. 1

Peng, B. and Zhang, F. Online prediction in sub-linear space. In Bansal, N. and Nagarajan, V. (eds.), *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pp. 1611–1634. SIAM, 2023. doi: 10.1137/1.9781611977554.ch60. URL https://doi.org/10.1137/1.9781611977554.ch60. 3

Radlinski, F., Kleinberg, R., and Joachims, T. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning*, pp. 784–791, 2008. 1

Robbins, H. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952. 1

Sauré, D. and Zeevi, A. Optimal dynamic assortment planning with demand learning. *Manuf. Serv. Oper. Manag.*, 15(3):387–404, 2013. doi: 10.1287/msom.2013.0429. URL https://doi.org/10.1287/msom.2013.0429. 1

Slivkins, A. Introduction to multi-armed bandits. *Found. Trends Mach. Learn.*, 12(1-2):1–286, 2019. doi: 10.1561/2200000068. URL https://doi.org/10.1561/2200000068. 1

Srinivas, V., Woodruff, D. P., Xu, Z., and Zhou, S. Memory bounds for the experts problem. In Leonardi, S. and Gupta, A. (eds.), *STOC '22: 54th Annual ACM SIGACT*

Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022, pp. 1158–1171. ACM, 2022. doi: 10.1145/3519935.3520069. URL https://doi.org/10.1145/3519935.3520069. 3

Tao, C., Zhang, Q., and Zhou, Y. Collaborative learning with limited interaction: Tight bounds for distributed exploration in multi-armed bandits. In *In FOCS 2019*, 2019. 3

Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933. 1

A Technical Preliminaries

We use the following standard variant of Chernoff-Hoeffding bound.

Proposition A.1 (Chernoff-Hoeffding bound)**.** *Let $X_1, \ldots, X_m$ be m independent random variables with support in $[0,1]$. Define $X := \sum_{i=1}^{m} X_i$. Then, for every $t > 0$,*

$$\Pr\left(|X - \mathbb{E}[X]| > t\right) \leq 2 \cdot \exp\left(-\frac{2t^2}{m}\right).$$

A direct corollary of this bound that we use in our proofs is the following.

Lemma A.2. *Let arm_1 and arm_2 be two different arms with rewards μ_1 and μ_2. Suppose we sample each arm $4 \cdot \frac{S}{\theta^2}$ times for some $S \geq 2$ to obtain empirical rewards $\widehat{\mu}_1$ and $\widehat{\mu}_2$. Then, if $\mu_1 - \mu_2 \geq c \cdot \theta$ for some integer $c \geq 1$, we have*

$$\Pr\left(\widehat{\mu}_1 \leq \widehat{\mu}_2\right) \leq \left(\frac{1}{2}\right)^{c^2-1} \cdot \exp\left(-S\right).$$

Proof. The proof is a standard application of the Chernoff bound Proposition A.1. For the empirical reward of $\widehat{\mu}_2$ to be greater than $\widehat{\mu}_1$, both of the low-probability following events are neceesary to happen:

$$\Pr\left(\widehat{\mu}_1 \leq \mu_1 - c \cdot \theta/2\right) \leq \exp\left(-2 \cdot (c \cdot \theta/2)^2 \cdot (4S/\theta^2)\right) \leq \exp\left(-c^2 \cdot S\right);$$
$$\Pr\left(\widehat{\mu}_2 \geq \mu_2 + c \cdot \theta/2\right) \leq \exp\left(-2 \cdot (c \cdot \theta/2)^2 \cdot (4S/\theta^2)\right) = \exp\left(-c^2 \cdot S\right).$$

For $c = 1$, a union bound on the events above gives us the desired bound. For $c \geq 2$, we have

$$\begin{aligned} \exp\left(-c^2 \cdot S\right) &\leq \exp\left(-c^2\right) \cdot \exp\left(-S\right) && (sc^2 \geq s + c^2 \text{ for } S \geq 1 \text{ and } c \geq 2) \\ &\leq \left(\frac{1}{2}\right)^{c^2} \cdot \exp\left(-S\right), \end{aligned}$$

and applying a union bound over the two cases gives us the desired statement. □

B Missing Details of Section 3

We start with introducing the guarantee of the ε-best arm algorithms in (Assadi & Wang, 2020; Jin et al., 2021).

Proposition B.1 ((Assadi & Wang, 2020; Jin et al., 2021))**.** *There exists a single-pass streaming algorithm that given a stream of stochastic multi-armed bandits, an error parameter $\varepsilon \in (0,1)$, and a confidence parameter $\delta \in (0,1)$, with probability at least $1-\delta$ return an arm with reward μ_ε such that $\mu_\varepsilon \geq \mu^* - \varepsilon$ with $O(\frac{K}{\varepsilon^2}\log(\frac{1}{\delta}))$ arm pulls and a memory of a single arm.*

For completeness, we include the algorithm of (Jin et al., 2021) that achieves the property described in Proposition B.1 – for our purpose, the algorithm of (Jin et al., 2021) is strictly better than that of (Assadi & Wang, 2020) since the latter needs a memory of 2 arms, and the sample complexity as an additive term proportional to $1/\varepsilon^3$. The algorithm of (Jin et al., 2021) can be described as follows.

Parameter Set 2:

$$\{s_\ell\}_{\ell \geq 0} : r_0 := 0, \qquad s_1 := \frac{16}{\varepsilon^2} \cdot \log(\frac{1}{\delta}) \qquad s_\ell := (2^\ell - 2^{\ell-1}) \cdot s_1 \qquad \text{(number of samples used in each level)}$$

$$\{\tau_j\}_{j \geq 1} : \tau_j := \frac{32}{\varepsilon^2} \cdot \log(\frac{j^2}{\delta}) \qquad \text{(the "total budget" threshold for comparing with the } j\text{-th arriving arm)}$$

$$p_j = \frac{1}{\log(j)+1} \qquad \text{(probability for setting the values of the gap parameter)}$$

The Single-pass ε-best Arm Algorithm of (Jin et al., 2021)

1. Maintain a stored arm arm^o and empirical reward $\widehat{\mu}^*$ of the stored arm.
2. After each update of arm^o, start an *epoch* as follows:
   (1) Let arm_j be the j-th arm after an epoch.
   (2) Sample $\alpha = \frac{\varepsilon}{4}$ with probability p_j and $\alpha = \frac{\varepsilon}{2}$ with probability $1 - p_j$.
   (3) Starting from level $\ell = 1$:
      i. Sample arm_j for s_ℓ times and get $\widehat{\mu}_{\mathsf{arm}_j}$.
      ii. If $\widehat{\mu}_{\mathsf{arm}_j} < \mu^*_\ell + \alpha$, drop arm_j;
      iii. Otherwise, if $2^\ell \cdot s_1 > \tau_j$, replace arm^o with arm_j and set $\mu^*_\ell = \widehat{\mu}_{\mathsf{arm}_j}$, and start a new epoch from Line 2.
      iv. Otherwise, send arm_j to the next level by calling Line 2((3))i with $(\ell = \ell + 1)$.
   (4) Output arm^o by the end of the stream.

It is easy to observe that the algorithm only uses a memory of a single arm (in addition to the one in the buffer). (Jin et al., 2021) proved that with high probability, the algorithm uses at most $O(\frac{K}{\varepsilon^2}\log(\frac{1}{\delta}))$ arm pulls and returns an ε-best arm. We now show that by picking the appropriate ε, it is straightforward to attain the $O(K^{1/3}T^{2/3})$ regret for any constant probability.

Proof of Theorem 2. The algorithm is simply as follows.

1. Run the algorithm in Proposition B.1 with parameter $\varepsilon = \frac{1}{2}\cdot(\frac{K}{T})^{1/3}$, obtain arm_ε.
2. Commit to arm_ε for all the remaining trials.

It is easy to see the algorithm only requires a single-arm memory. As such, we only need to analyze the regret. Note that the regret to find the ε-best arm is at most

$$\frac{2K}{(\frac{K}{T})^{2/3}}\cdot\log(\frac{1}{\delta}) = 2\log(1/\delta)\cdot K^{1/3}T^{2/3}.$$

On the other hand, conditioning on the algorithm succeeds, which happens with probability $1-\delta$, the reward gap between the best arm and the arm we commit to is at most $(\frac{K}{T})^{1/3}$. As such, the total regret is at most

$$T\cdot(\frac{K}{T})^{1/3} = K^{1/3}T^{2/3}.$$

Summing up the two regret terms gives us the desired statement. □

C Warm-up: A single-arm memory algorithm with $O(K^{1/3}T^{2/3}\log(K))$ expected regret

To begin with, we first give an algorithm with $O(K^{1/3}T^{2/3}\log(K))$ regret by analyzing the naive uniform elimination algorithm (folklore, see also (Even-Dar et al., 2002)) for ε-best arm. The algorithm is given as follows.

Naive Uniform Elimination – input parameters $\varepsilon \in (0,1)$, $\delta \in (0,1)$

1. Maintain space of a single extra arm and a best mean reward $\widehat{\mu}^*$ with initial value 0.
2. For each arriving arm_i pull $\frac{16}{\varepsilon^2}\log(\frac{K}{\delta})$ times, record the empirical reward $\widehat{\mu}_i$.
3. If $\widehat{\mu}_i > \widehat{\mu}^*$, discard the stored arm and let the arm_i be the stored arm; update $\widehat{\mu}^* = \widehat{\mu}_i$.

4. Otherwise, discard $\widehat{\mu}_i$ and keep the stored arm unchanged.
5. Return the stored arm by the end of the stream.

It is straightforward to see that the naive uniform elimination algorithm only requires a memory of a single-arm. Furthermore, the total number of arm pulls of the algorithm is clearly $\frac{16K}{\varepsilon^2}\log(\frac{K}{\delta})$. Note that the algorithm description is slightly different from the vanilla Uniform Elimination algorithm as described in (Even-Dar et al., 2002) – the importance of the subtle difference will be clear in the analysis, which we show as the follows.

Lemma C.1. *For fixed parameters $\delta \in (0,1)$, $\varepsilon \in (0,1)$, and integer $c \geq 1$, the Naive Uniform Elimination algorithm returns an $\overline{\mathsf{arm}}$ with reward*

$$\mu_{\overline{\mathsf{arm}}} \geq \mu^* - c \cdot \varepsilon$$

with probability at least $1 - (\frac{1}{2})^{c^2} \cdot \delta$.

Proof. The lemma is obtained by straightforward applications of the Chernoff bound and Lemma A.2. Concretely, the arm-pulling line in the Native Uniform Elimination algorithm is equivalent to setting $S = 4\log(\frac{K}{\delta})$ for each arm comparison in Lemma A.2. As such, for a fixed integer $c \geq 1$, when arm^* arrives, it has empirical reward

$$\begin{aligned}\Pr\left(\widehat{\mu}_{\mathsf{arm}^*} < \mu^* - c \cdot \frac{\varepsilon}{2}\right) &\leq (\frac{1}{2})^{c^2-1} \cdot \exp(-4\log(\frac{K}{\delta})) \\ &\leq (\frac{1}{2})^{c^2+1} \cdot \frac{\delta}{K}.\end{aligned}$$

As such, with probability at least $1 - (\frac{1}{2})^{c^2+1} \cdot \frac{\delta}{K}$, the estimation of $\widehat{\mu}^*$ eventually becomes at least $\mu^* - c \cdot \frac{\varepsilon}{2}$. On the other hand, if an arm_i has a reward less than $\mu^* - c \cdot \varepsilon$ we have

$$\begin{aligned}\Pr\left(\widehat{\mu}_i > \mu_i + c \cdot \frac{\varepsilon}{2}\right) &\leq (\frac{1}{2})^{c^2-1} \cdot \exp(-4\log(\frac{K}{\delta})) \\ &\leq (\frac{1}{2})^{c^2+1} \cdot \frac{\delta}{K}.\end{aligned}$$

And a union bound over at most K arms gives us that no arm with a mean reward less than $\mu^* - c \cdot \varepsilon$ can be stored in the end with probability at least $(\frac{1}{2})^{c^2+1} \cdot \delta$. Finally, we take a union bound over the failure probability of the aforementioned events, and conclude that with probability at least $1 - (\frac{1}{2})^{c^2} \cdot \delta$, the final returned arm is with a mean reward at least $\mu^* - c \cdot \varepsilon$. □

With Lemma C.1 establishing the 'smooth failure probability', we can now apply Lemma 4.1 to obtain the regret guarantee for streaming algorithms with uniform elimination.

Proposition C.2. *There exists a single-pass streaming algorithm that given a multi-armed bandit instance arriving in a stream with fixed parameters T, K such that $T > K$, carries out arm pulls with expected regret $\mathbb{E}[R_T] \leq O(K^{1/3}T^{2/3}\log(K))$ and uses a memory of a single extra arm.*

Proof. By Lemma C.1, we know that for any given parameter ε, there is

$$\Pr\left(\mu_{\overline{\mathsf{arm}}} < \mu^* - c \cdot \varepsilon\right) \leq (\frac{1}{2})^{c^2} \cdot \frac{1}{10} \leq (\frac{1}{2})^{c} \cdot \frac{1}{10}.$$

by setting $\delta = \frac{1}{10}$. As such, we can match the parameters in Lemma 4.1 by $S = 1$ and $M = 16K\log(10K)$. This gives us the desired bound of

$$\mathbb{E}[R_T] \leq O\left(M \cdot \frac{T^{2/3}}{K^{2/3}} + K^{1/3}T^{2/3}\right) = O(K^{1/3}T^{2/3}\log(K)).$$

□

D Missing Details of Section 5

We present the full simulation results and discussions in this section.

D.1 Simulation and Experiment Settings

We start with introducing the details for the experiment setup. We test the algorithms for arms with Bernoulli reward distributions. If the mean of the reward is μ, to simulate a pull of a Bernoulli arm, it suffices to draw a uniform at random sample from $[0, 1]$ and see if it is below μ [5]. We construct the stream of arms as a buffer, and the buffer can feed arms to the algorithm whenever needed. In particular, we test two types of streams:

1. The *uniform reward* setting: all the rewards of the arms are generated uniformly at random from $(0, 1)$.
2. The *standout* setting: there is one arm with mean reward $\mu = 0.82$, and all other arms are with mean reward μ drawn from a truncated Gaussian distribution with mean 0.5 and upper tail cutting-off at 0.8.

The stream is then ordered randomly by the buffer before it is fed into the algorithms.

We consider the number of arms with $K = 500$, $K = 5000$, and $K = 50000$. In each case, we further consider different number of arm pulls: $T = 1000K$, $T = 1000K^2$, and $T = 1000K^3$. In the implementation of different algorithms, we keep the leading constant to be 1 (i.e. we treat $O(\cdot)$ operation as with multiplicative factor of 1) except for multiplicative factor in the multi-level increment of samples (which we use 1.2 instead since it has to be > 1). We also keep the same ε across levels (as opposed to using $\frac{\varepsilon}{2^\ell}$) since the number of levels is small in our experiments. The simulations are all carried on a personal device with Apple M1 chip and 8GB memory, and each setting contains 50 runs with *fixed* random seeds from 0 to 49 for reproducibility.

D.2 Simulation Results

We report the simulation results for each *number of arms* and *type of stream* settings separately, and merge the other factors into separate tables and plots, respectively. The regrets in the tables are in the *relative* scale, i.e., we treat the regret of the uniform exploration algorithm as the benchmark (1.0), and compute the relative regrets of other algorithms.

Tables tables 2 to 1 summarize the mean and median regrets of the uniform reward setting; and Tables 5 to 7 give the mean and median regrets of the *standout* streaming setting, where there is an arm whose reward is much better than others.

Table 2. The comparison of the relative regret for different algorithms under setting $K = 500$ uniform stream setting.

	Uniform Exploration	Naive Elimination	$\log(K)$ ε-best	$\log\log(K)$ ε-best	$\log^*(K)$ ε-best	Game-of-Arms
Mean Regret						
$T = 1000K$	1.0	2.5732	0.5068	0.43331	0.6790	1.1131
$T = 1000K^2$	1.0	2.3139	0.4242	0.3670	0.6828	0.9793
$T = 1000K^3$	1.0	1.9321	0.8298	0.6504	0.6411	0.9693
Median Regret						
$T = 1000K$	1.0	2.5816	0.5000	0.4359	0.6757	1.1111
$T = 1000K^2$	1.0	2.3153	0.4023	0.3604	0.6095	0.9768
$T = 1000K^3$	1.0	2.106	0.3941	0.3403	0.5225	0.8701

[5] Due to limited computational power, when the number of arm pulls is large, e.g. $> 10^5$, we approximate the arm pull result by directly adding a Gaussian noise to μ.

Table 3. The comparison of the relative regret for different algorithms under setting $K = 5000$ uniform stream setting.

	Uniform Exploration	Naive Elimination	$\log(K)$ ε-best	$\log\log(K)$ ε-best	$\log^*(K)$ ε-best	Game-of-Arms
Mean Regret						
$T = 1000K$	1.0	3.3398	0.4650	0.4291	0.6328	1.0713
$T = 1000K^2$	1.0	3.1102	0.3840	0.4773	0.6728	0.9241
$T = 1000K^3$	1.0	2.1362	0.5515	0.4880	0.5639	1.0329
Median Regret						
$T = 1000K$	1.0	3.3684	0.4653	0.4293	0.6379	1.0718
$T = 1000K^2$	1.0	3.0815	0.3838	0.4285	0.6276	0.9236
$T = 1000K^3$	1.0	2.3269	0.4429	0.3438	0.4334	0.9293

Table 4. (Repeat from Section 5) The comparison of the relative regret for different algorithms under setting $K = 50000$ uniform stream setting.

	Uniform Exploration	Naive Elimination	$\log(K)$ ε-best	$\log\log(K)$ ε-best	$\log^*(K)$ ε-best	Game-of-Arms
Mean Regret						
$T = 1000K$	1.0	3.7355	0.4274	0.4000	0.6012	1.0290
$T = 1000K^2$	1.0	3.0423	0.5989	0.4374	0.5733	1.2976
$T = 1000K^3$	1.0	2.8652	0.5686	0.5117	0.5982	1.0960
Median Regret						
$T = 1000K$	1.0	3.7555	0.4264	0.3994	0.6036	1.0008
$T = 1000K^2$	1.0	3.1974	0.5525	0.3776	0.5713	1.1953
$T = 1000K^3$	1.0	3.0008	0.4393	0.3789	0.5142	0.9996

Table 5. The comparison of the relative regret for different algorithms under setting $K = 500$ standout stream setting.

	Uniform Exploration	Naive Elimination	$\log(K)$ ε-best	$\log\log(K)$ ε-best	$\log^*(K)$ ε-best	Game-of-Arms
Mean Regret						
$T = 1000K$	1.0	2.4357	0.6090	0.5606	0.7883	1.2690
$T = 1000K^2$	1.0	2.3154	1.0294	0.5368	0.7982	0.9398
$T = 1000K^3$	1.0	2.1100	1.9985	0.3553	1.7054	0.8556
Median Regret						
$T = 1000K$	1.0	2.6234	0.4762	0.4493	0.7530	1.1208
$T = 1000K^2$	1.0	2.3154	0.4195	0.3915	0.6476	0.9425
$T = 1000K^3$	1.0	2.1100	0.3778	0.3545	0.5900	0.8595

Table 6. The comparison of the relative regret for different algorithms under setting $K = 5000$ standout stream setting.

	Uniform Exploration	Naive Elimination	$\log(K)$ ε-best	$\log\log(K)$ ε-best	$\log^*(K)$ ε-best	Game-of-Arms
Mean Regret						
$T = 1000K$	1.0	2.9354	0.7552	0.6405	0.7705	1.2104
$T = 1000K^2$	1.0	2.9551	0.8746	0.4686	0.6112	0.8653
$T = 1000K^3$	1.0	2.6700	9.1241	4.3835	1.7245	0.7826
Median Regret						
$T = 1000K$	1.0	3.1183	0.8236	0.6978	0.8849	1.2310
$T = 1000K^2$	1.0	2.9551	0.3994	0.3728	0.6122	0.8656
$T = 1000K^3$	1.0	2.6700	0.3622	0.3392	0.5508	0.7835

Table 7. The comparison of the relative regret for different algorithms under setting $K = 50000$ standout stream setting.

	Uniform Exploration	Naive Elimination	$\log(K)$ ε-best	$\log\log(K)$ ε-best	$\log^*(K)$ ε-best	Game-of-Arms
Mean Regret						
$T = 1000K$	1.0	3.0179	0.6323	0.5341	0.7001	1.0888
$T = 1000K^2$	1.0	3.5402	0.8447	0.5144	0.5750	1.5099
$T = 1000K^3$	1.0	3.1806	20.4462	0.3182	0.5162	0.7344
Median Regret						
$T = 1000K$	1.0	3.0429	0.6213	0.5421	0.7270	1.0658
$T = 1000K^2$	1.0	3.5402	0.3780	0.3542	0.5752	0.8194
$T = 1000K^3$	1.0	3.1806	0.3398	0.3185	0.5156	0.7337

From the tables, it can be observed that the ε-best arm-based algorithms consistent outperform the benchmark uniform exploration. The naive elimination algorithm, on the other hand, offers generally poor performances – since we only test the number of trials for as large as $1000K^3$ due to limited computational resource, the term $(\log(T))^{1/3}$ is still much smaller than $\log(K)$. Testing trials with even larger scale will probably help the naive elimination algorithm to catch up in the performance.

Among the ε-best arm algorithms, it appears that the $\log\log(K)$-space algorithm consistently achieve the best mean and median regrets. The $\log(K)$-space algorithm is somehow unstable and offers much worse mean regret in the $K = 5000$ and $K = 50000$ standalone stream settings. It nonetheless consistently achieves much better median regrets. We suspect this is due to the success probability not sufficiently high, and the algorithm sometimes fails to capture an ε-best arm and commit all remaining trials to a 'wrong' arm[6]. This also explains why the performance of the $\log(K)$-space algorithm does *not* become worse in the uniform mean-reward setting. Interestingly, the GAME-OF-ARMS algorithm in (Assadi & Wang, 2020) offers better performance than the naive elimination algorithm, although theoretically, there is a $\frac{\log(K)}{\varepsilon^3}$ term on the sample complexity, which translates into $(T/K) \cdot \log(K)$ regret – a worse regret bound when T is very large.

We further provides figures of the regrets with the error bars, showing the fluctuations of regrets in each setting in more details. Since there are some huge gaps between the regrets with different algorithms, we use $\log_{10}(\cdot)$ scale for the regret.

From the figures, it can be observed that the $\log(K)$-space algorithm gives the most unstable performances and the most extreme outliers, while other ε-best algorithms are generally stable. In the uniform reward setting, the first and third quartiles of the rewards do not change drastically w.r.t. T, and there are generally less extreme outliers when T is larger. This is because in the uniform reward setting, the differences between the ε-best arms starts to matter, yet the cost of committing to a mediocre arm becomes lower. On the other hand, in the standout reward setting, when T is smaller, the first and third quartiles of the reward distributions have larger ranges, but there are generally less extreme outliers. This matches our understanding of the behaviors of the algorithms: when T is smaller, there a good chance that the algorithm terminates before finding an ε-best arm; on the other hand, when T becomes larger, committing to a 'wrong' arm is much more expensive.

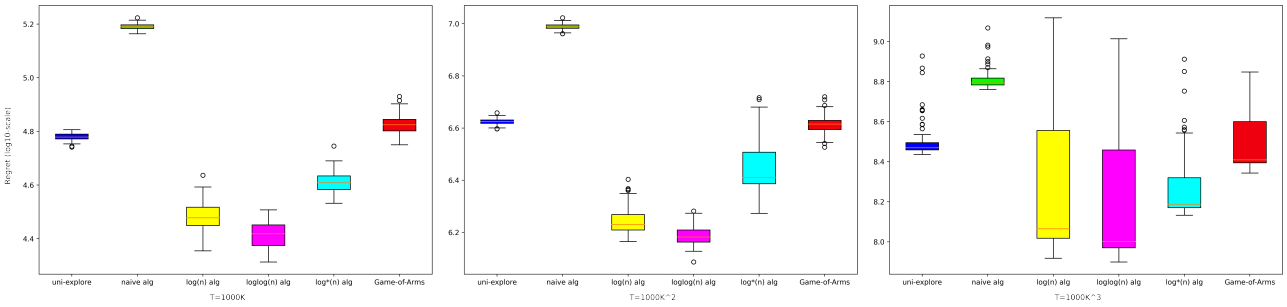

Figure 1. The regret error bars for $K = 500$ uniform reward setting of the stream.

[6]It is likely that this problem can be fixed by a heuristic search for the constant on the $\log(K)$-space algorithm. However, we do not pursue this direction in this paper.

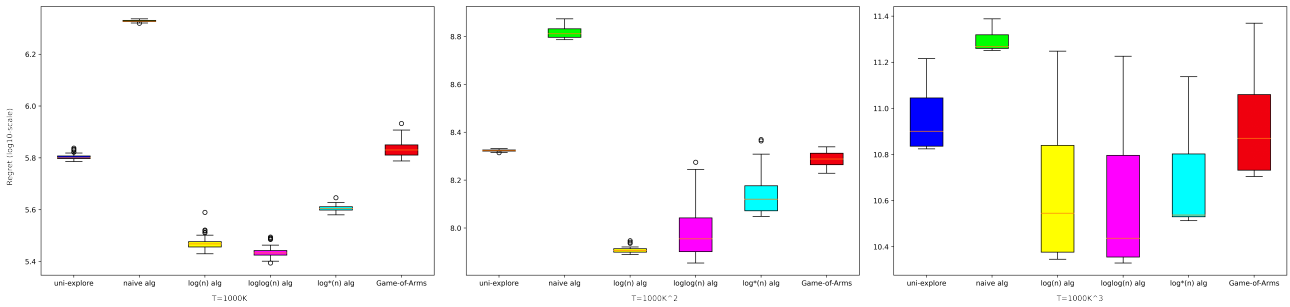

Figure 2. The regret error bars for $K = 5000$ uniform reward setting of the stream.

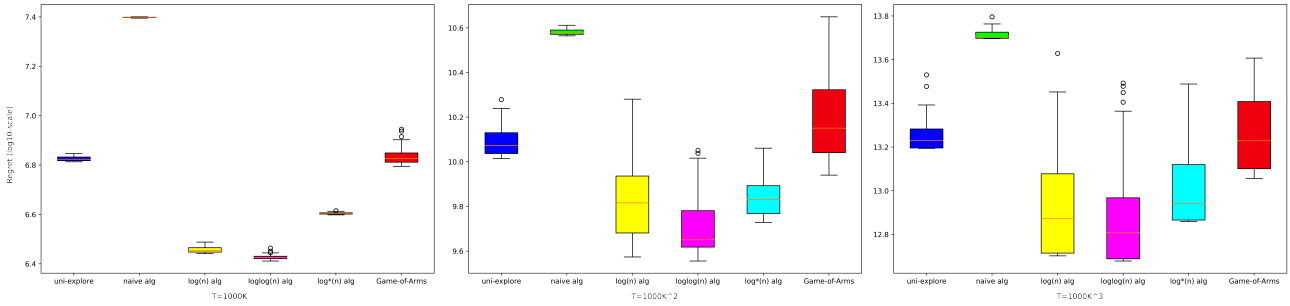

Figure 1. (Repeat from Section 5) The regret error bars for $K = 50000$ uniform reward setting of the stream.

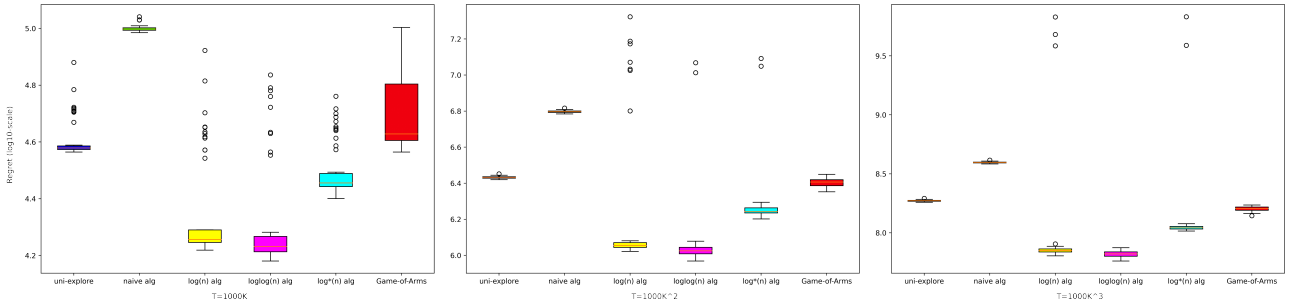

Figure 3. The regret error bars for $K = 500$ standout reward setting of the stream.

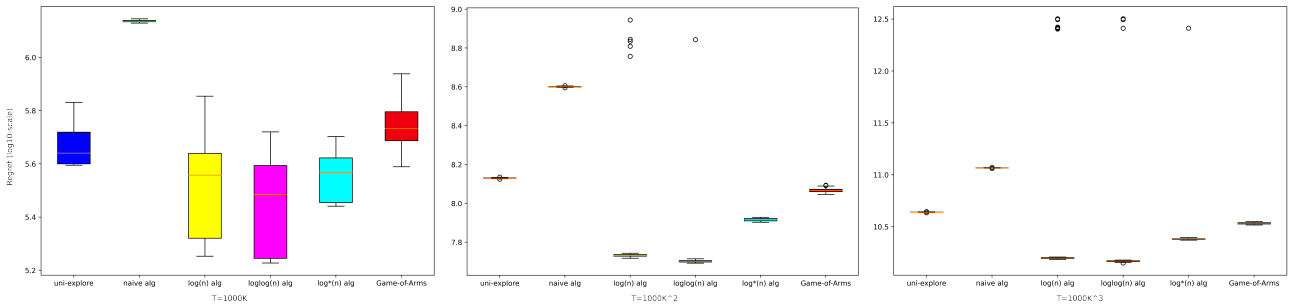

Figure 4. The regret error bars for $K = 5000$ standout reward setting of the stream.

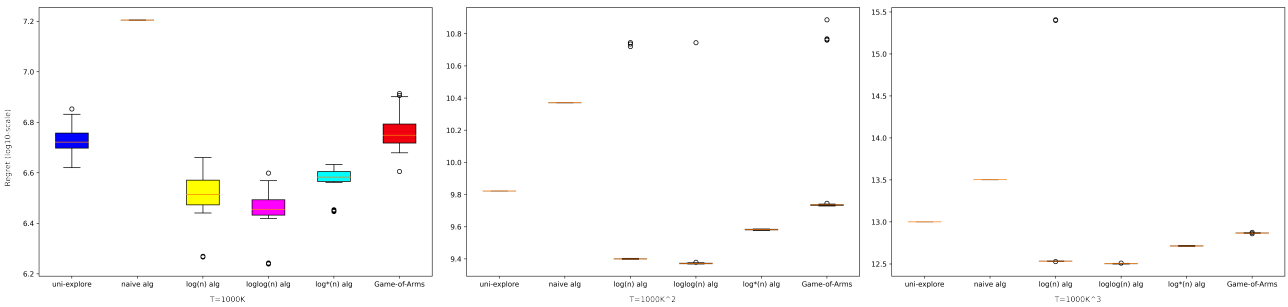

Figure 5. The regret error bars for $K = 50000$ standout reward setting of the stream.

E Details for Additional Algorithms

We provide the descriptions for the streaming implementation of the uniform exploration algorithm and the additional algorithms we mentioned in Remark 2.

E.1 Uniform Sampling Algorithm under the Streaming Setting

We first give the streaming implementation of simple uniform exploration algorithm. The algorithm is to simply pull each arm N times, pick the arm with the highest empirical reward, and commit to the returned arm for the rest of the trials (if any). As such, the streaming adaptation is extremely straightforward:

Streaming Uniform Exploration – parameters N: number of arm pulls for each arm

1. Maintain space of a single extra arm and a best mean reward $\widehat{\mu}^*$ with initial value 0.
2. For each arriving arm_i pull N times, record the empirical reward $\widehat{\mu}_i$.
3. If $\widehat{\mu}_i > \widehat{\mu}^*$, discard the stored arm and let the arm_i be the stored arm; update $\widehat{\mu}^* = \widehat{\mu}_i$.
4. Otherwise, discard $\widehat{\mu}_i$ and keep the stored arm unchanged.
5. Return the stored arm by the end of the stream.

It is easy to see that we only need to main a single arm (in addition to the arriving buffer) during the stream. Furthermore, it is folklore that if we set $N = O((\frac{T}{K})^{2/3}\log^{1/3}(T))$, the expected regret is attained at $O(K^{1/3}T^{2/3}\log^{1/3}(T))$.

E.2 The $\log(K)$- and $\log\log(K)$-memory streaming algorithms

We now introduce the algorithms used with $\log(K)$- and $\log\log(K)$-memory, which are implemented in Section 5 and at times offer more competitive performances than the $\log^*(K)$-memory algorithm. We opt to include their descriptions since the ε-best algorithms were not described in (Assadi & Wang, 2020). We however omit the proofs for the algorithms to achieve the optimal regret since it is very similar to Lemma 4.3, and leave it as an exercise for keen readers. For the $O(1)$-memory GAME-OF-ARMS algorithm and the single-arm memory algorithm, we refer the reader to the respective work (Assadi & Wang, 2020; Jin et al., 2021).

The algorithm with $O(\log(K))$ can be described as follows.

An algorithm with $O(\log K)$-arm space and $O(K^{1/3}T^{2/3})$ expected regret:

Input parameter: K number of arms; T number of trials

Parameters:

$$\varepsilon = (\frac{K}{T})^{1/3}$$
$$\varepsilon_\ell = \frac{1}{10}\cdot\frac{\varepsilon}{2^{\ell-1}}$$
$$\{s_\ell\}_{\ell\geq 1}: \quad s_\ell = \frac{4}{\varepsilon_\ell^2}\cdot\left(\ln(1/\delta) + 3^\ell\right).$$

Maintain:

- Buckets: B_1, B_2, ..., B_t, each of size 4 for $t := \lceil \log_4(K) \rceil$.

Algorithmic procedure:

- For each arriving arm_i in the stream do:

(1) Add arm_i to bucket B_1.
(2) If any bucket B_ℓ is full:
   (a) We sample each arm in B_ℓ for s_ℓ times;
   (b) Send the arm^*_ℓ with the highest empirical reward to $B_{\ell+1}$, an clear the bucket B_ℓ;

- At the end of the stream, pick the best arm of each bucket with s_ℓ times, repeat line (2) regardless of whether the bucket is full.
- Pick arm^*_t of bucket B_t as the selected arm, and commit the rest of the trials to this arm.

This algorithm is very similar to the $O(\log(K))$-arm algorithm for best-arm identification in (Assadi & Wang, 2020); in fact, the only technical difference between this algorithm and the original is the usage of exponentially decreasing ε across levels. We shall note that this is in contrast with the $O(\log\log(K))$-arm and $O(\log^*(K))$-arm algorithms, in which the modification to challenge a *fixed* reward threshold plays an important role. The algorithm with $O(\log\log(K))$-arm space can be shown as follows.

An algorithm with $O(\log\log K)$-arm space and $O(K^{1/3}T^{2/3})$ expected regret:

Input parameter: K number of arms; T number of trials

Parameters:

$$\varepsilon = (\frac{K}{T})^{1/3}$$
$$\varepsilon_\ell = \frac{1}{10}\cdot\frac{\varepsilon}{2^{\ell-1}}$$
$$\{s_\ell\}_{\ell\geq 1}: \quad s_\ell = \frac{4}{\varepsilon_\ell^2}\cdot\left(\ln(1/\delta)+3^\ell\right); \qquad s_T := \frac{4}{\varepsilon^2}\cdot(\ln(1/\delta)+\ln(K)).$$

Maintain:

- Buckets: B_1, B_2, ..., B_{t-1}, each of size 4 for $t := \lceil \log_4 \ln(K)\rceil$; B_t is of size 1.
- Best-reward on level t: μ^*_t initialized to 0.

Algorithmic procedure:

- For each arriving arm_i in the stream do:
  (1) Add arm_i to bucket B_1.
  (2) For any level $\ell < t$: B_ℓ is full:
     (a) We sample each arm in B_ℓ for s_ℓ times;
     (b) Send the arm^*_ℓ with the highest empirical reward to $B_{\ell+1}$, an clear the bucket B_ℓ;
  (3) For level t:
     (a) Sample the most recent arm that reaches level t s_t times, reward the empirical reward $\tilde{\mu}$;
     (b) If $\tilde{\mu} > \mu^*_t$, let the most recent arm be stored, discard the stored arm at level t, and update $\mu^*_t \leftarrow \tilde{\mu}$;
- At the end of the stream, pick the best arm of each bucket with s_ℓ times, and send the best to higher levels even regardless of whether the bucket is full.
- Pick the single arm^*_t of bucket B_t as the selected arm, and commit the rest of the trials to this arm.

Note that compared to the $O(\log\log(K))$-arm memory algorithm for best-identification in (Assadi & Wang, 2020), the small yet subtle difference here is that allow more 'slack' for the stored arm at level t by not repetitively pulling it and using the fixed reward threshold instead.