Efficient Algorithms for Extreme Bandits

Dorian Baudry, Yoan Russac, Emilie Kaufmann

Efficient Algorithms for Extreme Bandits

Dorian Baudry[1] **Yoan Russac**[2] **Emilie Kaufmann**[1]
1. CNRS, Univ. Lille, Inria, Centrale Lille, UMR 9189 - CRIStAL, F-59000, Lille, France
2. DI ENS, CNRS, ENS, Université PSL, Paris, France

Abstract

In this paper, we contribute to the Extreme Bandit problem, a variant of Multi-Armed Bandits in which the learner seeks to collect the largest possible reward. We first study the concentration of the maximum of i.i.d random variables under mild assumptions on the tail of the rewards distributions. This analysis motivates the introduction of Quantile of Maxima (QoMax). The properties of QoMax are sufficient to build an Explore-Then-Commit (ETC) strategy, QoMax-ETC, achieving strong asymptotic guarantees despite its simplicity. We then propose and analyze a more adaptive, anytime algorithm, QoMax-SDA, which combines QoMax with a subsampling method recently introduced by Baudry et al. (2021). Both algorithms are more efficient than existing approaches in two aspects (1) they lead to better empirical performance (2) they enjoy a significant reduction of the memory and time complexities.

1 INTRODUCTION

Multi-Armed Bandits (MAB) provide a powerful framework for balancing exploration and exploitation in sequential decision making tasks. In a MAB model, a learner is interacting with K unknown distributions (called arms) generating rewards, that we denote by $\nu_1, \dots, \nu_K$. In the most classical problem formulation, the learner sequentially samples the arms in order to maximize her expected sum of rewards. In this paper, we consider a different setting in which the learner seeks to collect the largest possible reward. This problem, first introduced by Cicirello and Smith (2005), is referred to as *Extreme Bandits* or *max K-armed bandit*. Obtaining the largest possible reward can be of interest for practical scenarios including financial (Gilli et al., 2006), medical (Neill and Cooper, 2010), online marketing (Skiera et al., 2010) applications.

Letting $X_{k,t}$ be the reward obtained from arm k at time t, a bandit algorithm (or policy) selects an arm I_t using past observations and receives the reward $X_{I_t,t}$. The rewards stream $(X_{k,t})$ is drawn i.i.d. from ν_k and independently from other rewards streams. In this work, we assume that all arms have an unbounded support (the finite support case is studied by Nishihara et al. (2016)). In this context, Carpentier and Valko (2014) define the extreme regret of a policy as

$$\mathcal{R}_T^\pi = \max_{k\leq K} \mathbb{E}[\max_{t\leq T} X_{k,t}] - \mathbb{E}_\pi[\max_{t\leq T} X_{I_t,t}] \,. \tag{1}$$

Two types of performance guarantees have been derived in previous works. Using the terminology of Bhatt et al. (2021), we say that π has a vanishing regret in the *weak* sense if

$$\mathcal{R}_T^\pi = \underset{T\to\infty}{o}\left(\max_{k\leq K} \mathbb{E}[\max_{t\leq T} X_{k,t}]\right) \tag{2}$$

and π has a vanishing regret in the *strong* sense if

$$\lim_{T\to\infty} \mathcal{R}_T^\pi = 0 \,. \tag{3}$$

While classical bandit algorithms aim for the arm with the largest expected reward, a good algorithm for extreme bandit should intuitively discover the arm with the heaviest tail. Existing algorithms for this problem can be divided into three categories: (1) Fully-parametric approaches (Cicirello and Smith, 2005; Streeter and Smith, 2006a) where the distributions are assumed to be known (Frechet, Gumbel). (2) Semi-parametric approaches (Carpentier and Valko, 2014; Achab et al., 2017) where distributions satisfy a second-order Pareto assumption. In Carpentier and Valko (2014), weak vanishing regret is obtained for second-order Pareto distributions assuming that a lower bound on a parameter of the distribution is known to the algorithm. Achab et al. (2017) refine

Proceedings of the 25[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

this analysis and obtain strong vanishing regret when this lower-bound is large enough. (3) Distribution-free approaches (Streeter and Smith, 2006b; Bhatt et al., 2021) which do not leverage any assumption on the reward distributions. A simple algorithm, ThresholdAscent, was proposed in Streeter and Smith (2006b), but without theoretical guarantees. Bhatt et al. (2021) recently proposed Max-Median, an algorithm based on robust statistics that can be employed for any kind of distribution. Max-Median is proved to have weak vanishing regret for polynomial-like arms and strongly vanishing regret for exponential-like arms.

In this work, we revisit the extreme bandit problem with the idea of designing algorithms based on *pairwise comparisons of tails* with provable guarantees under minimal assumptions on the arms. The motivation stems from a recent line of work on subsampling algorithms for classical bandits (Baudry et al., 2020) which performs "fair" pairwise comparisons of empirical means based on an equal sample size and attains good performance for several types of distributions.

In Section 2, we highlight the limitation of comparing directly the maxima of n i.i.d. samples and introduce the *Quantile of Maxima* (QoMax) estimator. Instead of computing the maximum of n samples, the learner separates the collected data into *batches* of equal size and compute the quantile of order q of the maxima over the different batches. QoMax is inspired by the Median of Means estimator (Alon et al., 1999) that was used for heavy-tail bandits (Bubeck et al., 2013). We derive upper bounds on the probability that one QoMax exceeds another, that are instrumental to design our algorithms. In Section 3, we first propose an Explore-Then-Commit algorithm using QoMax, for which we establish vanishing regret in the strong sense under the mild assumption that the bandit model has a dominant arm. Albeit simple, this approach requires some tuning which depends on the horizon T. To overcome this limitation, we propose in Section 4 the QoMax-SDA algorithm which combines QoMax with the subsampling strategy from Baudry et al. (2021). We prove that it achieves vanishing regret for arms with exponential or polynomial tails and also provide some elements of analysis under the weaker dominant arm assumption. In Section 5, we highlight the efficiency of our algorithms which allow for a significant reduction of the storage and computational cost while outperforming existing approaches empirically.

2 COMPARING TAILS

In this section, we motivate our new QoMax estimator used for comparing the tails of two distributions based on n i.i.d. samples of each. We first present the assumptions under which we are able to analyze QoMax and the resulting extreme bandit algorithms.

We define the *survival function* G of a distribution ν as $G(x) = \mathbb{P}_{X\sim\nu}(X > x)$ for all $x \in \mathbb{R}$. We shall consider two different assumptions for arms' distributions.

Definition 1 (Exponential or polynomial tails)**.** *Let ν be a distribution of survival function G. **(1)** If there exists $C > 0$ and $\lambda > 1$ such that $G(x) \sim Cx^{-\lambda}$ we say that ν has a **polynomial tail**. **(2)** If there exists $C > 0, \lambda \in \mathbb{R}^+$ such that $G(x) \sim C\exp(-\lambda x)$ we say that ν has an **exponential tail**.*

These *semi-parametric* assumptions (which says nothing about the lower part of the distribution) have been introduced by Bhatt et al. (2021). We remark that a polynomial tail is a weaker condition than the second-order Pareto assumption from Carpentier and Valko (2014). Now, we introduce a general assumption which allows to compare two (arbitrary) tails.

Definition 2 (Dominating tail)**.** *Let G_1 and G_2 be the survival functions of two distributions ν_1 and ν_2. We say that the tail of ν_1 **dominates** the tail of ν_2 (we write $\nu_1 \succ \nu_2$) if there exists $C > 1$ and $x \in \mathbb{R}$ such that for all $y > x$, $G_1(y) > CG_2(y)$.*

In the rest of the paper, we will consider a bandit model that has a dominating arm, denoted by 1 without loss of generality: $\nu_1 \succ \nu_k$ for all $k \neq 1$. Under this assumption, arm 1 is optimal in the sense that for T large enough an oracle strategy would select this arm only. To the best of our knowledge, this is the weakest assumption introduced so far for extreme bandits.

2.1 Comparing Maxima

Let ν_1 and ν_2 be two distributions from which we observe n i.i.d. samples denoted by $X_{1,1}, \dots, X_{1,n}$ and $X_{2,1}, \dots, X_{2,n}$ respectively. A natural idea to compare their tails is to use the samples' maxima, $X^+_{k,n} = \max\{X_{k,1}, \dots, X_{k,n}\}$ for $k \in \{1, 2\}$. For these estimators to serve as a proxy for comparing the tails, we need the probability $\mathbb{P}(X^+_{1,n} < X^+_{2,n})$ to decay fast enough when $\nu_1 \succ \nu_2$. To upper bound this probability, we note that for any sequence (x_n),

$$\mathbb{P}(X^+_{1,n} < X^+_{2,n}) \leq \mathbb{P}(X^+_{1,n} \leq x_n) + \mathbb{P}(X^+_{2,n} > x_n) \,.$$

Using first that $\mathbb{P}(X^+_{1,n} \leq x) \leq \exp(-nG_1(x))$ and then $\mathbb{P}(X^+_{2,n} > x) \leq nG_2(x)$, and optimizing for x_n yields the following result, proved in Appendix A.

Lemma 1 (Comparison of Maxima)**.** *Assume that both ν_1 and ν_2 have either polynomial or exponential tails, with respective second parameter λ_1 and λ_2, with $\lambda_1 < \lambda_2$ (so that $\nu_1 \succ \nu_2$). Define $\delta = \frac{\lambda_2}{\lambda_1} - 1 > 0$,*

then there exists a sequence (x_n) *such that*

$$\max\{\mathbb{P}(X^+_{1,n} \leq x_n), \mathbb{P}(X^+_{2,n} \geq x_n)\} = \mathcal{O}\left(\frac{(\log n)^{\delta+1}}{n^\delta}\right).$$

Lemma 1 shows that even under the stronger semi-parametric assumption, $\mathbb{P}(X^+_{1,n} \leq X^+_{2,n})$ does not decay exponentially fast, unlike what happens when we compare the empirical means of light-tailed distributions. Furthermore, the rate δ is problem-dependent and can be arbitrarily small. As pointed out by Carpentier and Valko (2014) it can actually be seen as the Extreme Bandits equivalent of the *gap* in bandits, we therefore call δ the ***tail gap***. Besides, we prove in Lemma 4 (in Appendix) a lower bound of order $\mathcal{O}(n^{-(1+\delta)})$, which motivates using more robust statistics on the distributions' tails.

2.2 Quantile of Maxima (QoMax)

Results similar to those of Section 2.1 have been previously encountered in the bandit literature. In (Bubeck et al., 2013), the authors study the problem of bandit with heavy tails, prove a concentration inequality in $n^{-\delta}$ for some $\delta > 0$ and use this result to build several estimators with faster convergence. Among them, they consider the Median-of-Means (MoM) introduced by Alon et al. (1999). We build a natural variant of MoM, that we call Quantile of Maxima (QoMax). The principle of QoMax is simple: the learner chooses a quantile q, and has access to $N = b \times n$ data $\mathcal{X} = (X_{m,i})_{m\leq n, i\leq b}$. It then allocates the data in b batches of size n and: (1) find the maximum of each batch, (2) compute the quantile of order q over the b maxima. We summarize QoMax in Algorithm 1.

Algorithm 1 Quantile of Maxima (QoMax)

Input: quantile q, b batches of size n, observations $(X_{m,i})_{m\leq n, i\leq b}$
for $i = 1, \dots, b$ **do**
  Compute $(X^+_n)^{(i)} = \max\{X_{1,i}, \dots, X_{n,i}\}$
Return: quantile of order q of $\{(X^+_n)^{(1)}, \dots, (X^+_n)^{(b)}\}$

For a finite set of size b, we simply define the quantile q as the observation of rank $\lceil bq \rceil$ in the list of sorted data (in increasing order). In the sequel we denote by $\bar{X}^q_{k,n,b}$ the QoMax of order q computed from b batches of size n of i.i.d. replications from arm k.

We are now ready to state the crucial property of QoMax estimators that will be used in our two analyses.

Theorem 1 (Comparison of QoMax). *Let* ν_1 *and* ν_2 *be two distributions satisfying* $\nu_1 \succ \nu_2$ *and* $q \in (0,1)$. *Then,* ***there exists*** *a sequence* x_n, *a constant* $c > 0$, *and an integer* $n_{\nu_1,\nu_2,q}$ *such that for* $n \geq n_{\nu_1,\nu_2,q}$,

$$\max\left\{\mathbb{P}(\bar{X}^q_{1,n,b} \leq x_n), \mathbb{P}(\bar{X}^q_{2,n,b} \geq x_n)\right\} \leq \exp(-cb)\,.$$

If the tails are furthermore either polynomial or exponential with a ***positive tail gap****, then the result holds* ***for any*** $c > 0$ *and* n *larger than some* $n_{c,\nu_1,\nu_2,q}$.

It follows from Theorem 1 that $\mathbb{P}(\bar{X}^q_{1,n,b} \leq \bar{X}^q_{2,n,b}) \leq 2\exp(-cb)$ for n large enough. Strikingly, this result tells us that, under the simple assumption that one tail dominates the comparison of QoMax computed with the same parameters will not be in favor of the dominating arm with a probability that **decreases exponentially with the batch size**.

Remark 1. *In general QoMax is* ***not*** *an estimate of the expectation of the maximum. We will use it to* ***compare two tails****, in order to find the heavier.*

Remark 2 (Choice of quantile level q)**.** *Note that Theorem 1 holds for any value of* $q \in (0,1)$, *but the impact of* q *is materialized in the (problem-dependent) sample size* $n_{\nu_1,\nu_2,q}$ *needed for the inequality to hold. For the practitioner, we think that in most cases choosing* $q = 1/2$ *is appropriate. Still, in Section 5 we exhibit a difficult setting where a choice of* q *close to* 1 *is helpful.*

2.3 Proof of Theorem 1

We let $\mathrm{kl}(x,y) = x\log(x/y) + (1-x)\log((1-x)/(1-y))$ denote the binary relative entropy. Just like for the analysis of Median-of-Means, the starting point is to relate deviations inequalities for a QoMax to deviation inequalities for binomial distributions. Letting $(X^+_{1,n})^{(i)}$ (resp. $(X^+_{2,n})^{(i)}$) denote the maximum over the i-th batch of observations from ν_1 (resp. ν_2),

$$\begin{aligned}\mathbb{P}(\bar{X}^q_{1,n,b} \leq x) &\leq \mathbb{P}\left(\sum_{i=1}^b \mathbb{1}\left((X^+_{1,n})^{(i)} \leq x\right) \geq bq\right) \\ &\leq \exp(-b \times \mathrm{kl}(q, \mathbb{P}(X^+_{1,n} \leq x)))\,.\end{aligned}$$

The last step applies the Chernoff inequality to a binomial distribution with parameters b and $p = \mathbb{P}(X^+_{1,n} \leq x)$, and holds whenever $\mathbb{P}(X^+_{1,n} \leq x) \leq q$. Similarly, if $\mathbb{P}(X^+_{2,n} \geq x) \leq 1 - q - 1/b$, we have

$$\begin{aligned}\mathbb{P}(\bar{X}^q_{2,n,b} \geq x) &\leq \mathbb{P}\left(\sum_{i=1}^b \mathbb{1}\left((X^+_{2,n})^{(i)} \geq x\right) \geq b - bq - 1\right) \\ &\leq \exp(-b\mathrm{kl}(1-q-1/b, \mathbb{P}(X^+_{2,n} \geq x)))\end{aligned}$$

For exponential and polynomial tails, thanks to Lemma 1 there exists a sequence (x_n) such that both

$\mathbb{P}(X^+_{1,n} \leq x_n)$ and $\mathbb{P}(X^+_{2,n} \geq x_n)$ converge to zero, and the result follows easily. Under the dominance assumption, the following result controls the deviations of the maxima and is proved in Appendix A.

Lemma 2. *Assume that $\nu_1 \succ \nu_2$. Then, for any $q \in (0,1)$ there exists $n_{\nu_1,\nu_2,q} \in \mathbb{N}$, a sequence x_n and some $\varepsilon > 0$ such that for all $n \geq n_{\nu_1,\nu_2,q}$ and b large enough,*

$$\mathbb{P}(X^+_{1,n} \leq x_n) \leq q - \varepsilon \ , \quad \text{and} \quad \mathbb{P}(X^+_{2,n} \leq x_n) \geq q + \varepsilon \ .$$

With the notation of Lemma 2, Theorem 1 then holds for $c = \min\left(\mathrm{kl}(q, q-\varepsilon), \mathrm{kl}(1-q-\varepsilon/2, 1-q-\varepsilon)\right)$ provided that the batch size is larger than $2/\varepsilon$.

3 QoMax-ETC

In this section, we propose QoMax-ETC, a simple Explore-Then-Commit algorithm using QoMax estimators. The algorithm is reported in Algorithm 2 and works as follows. First, the learner selects a quantile q, and given the time horizon T picks a batch size b_T and a sample size n_T. Then, the exploration phase starts where every arm is pulled $N_T = b_T \times n_T$ times allocated in b_T batches of size n_T. At the end of this step, the learner computes a q-QoMax estimator from the history of each arm using the different batches. Next comes the exploitation phase where the algorithm pulls the arm I_T with the largest QoMax until time T.

Algorithm 2 QoMax-ETC

Input: K arms, horizon T, quantile q, number of batches b_T, number of samples per batch n_T

for $k = 1, \dots, K$ **do**
  Pull arm k, $b_T \times n_T$ times
  Allocate the data in b_T batches of size n_T
  Compute their QoMax, $\bar{X}^q_{k,n_T,b_T}$ (Algorithm 1)

for $t = K \times n_T \times b_T + 1, \dots, T$ **do**
  Pull arm $I_T = \mathrm{argmax}_k \bar{X}^q_{k,n_T,b_T}$

We remark that an ETC algorithm has already been proposed by Achab et al. (2017) for extreme bandits. Their algorithm differs from ours by the choice of the arm I_T drawn in the exploitation phase: they build an upper confidence bound on the maximum under the assumption that the distributions are second-order Pareto and select I_T as the arm with largest upper confidence bound. In contrast, QoMax-ETC does not assume anything about the arms distributions.

We now analyze QoMax-ETC under a bandit model $\nu = (\nu_1, \dots, \nu_K)$ such that $\nu_1 \succ \nu_k$ for all $k \neq 1$.

Proposition 1 (Regret of QoMax-ETC)**.** *Let π be an ETC policy sampling $N_T = n_T \times b_T$ times each arm during the exploration phase. If $T \geq KN_T$,*

$$\mathcal{R}^\pi_T \leq \underbrace{\mathbb{E}\left[\max_{t\leq T} X_{1,t}\right] - \mathbb{E}\left[\max_{t\leq T-KN_T} X_{1,t}\right]}_{\textit{Exploration cost}} + \underbrace{\mathbb{P}(I_T \neq 1)\mathbb{E}\left[\max_{t\leq T} X_{1,t}\right]}_{\textit{Cost of picking a wrong arm}} .$$

We prove this result in Appendix B. This proposition shows that the regret of the ETC algorithm can be properly controlled by two factors **(1)** the probability of picking a wrong arm for the exploitation phase, **(2)** the gap between the growth rate of the maximum over T or $T - KN_T$ observations of the dominant arm, that we call "exploration cost" as it is fully determined by the length of the exploration phase and the arms' distributions. In the rest of the paper we will assume that the distribution of the dominant arm satisfies the following assumption.

Assumption 1. *$\mathbb{E}\left[\max_{t\leq T} X_{1,t}\right] = o(T)$, and for any $\gamma < 1$ if $N_T = o(T^\gamma)$ then*

$$\mathbb{E}\left[\max_{t\leq T} X_{1,t}\right] - \mathbb{E}\left[\max_{t\leq T-KN_T} X_{1,t}\right] \xrightarrow[T\to+\infty]{} 0 \ .$$

This condition is satisfied for nearly all distributions encountered in practice (e.g polynomial, exponential or gaussian tails) as discussed in Appendix A, in which we provide explicit upper bounds on the exploration cost. We now state our main theoretical claim for QoMax-ETC.

Theorem 2 (Vanishing regret of QoMax-ETC)**.** *Consider a bandit $\nu = (\nu_1, \dots, \nu_K)$ with $\nu_1 \succ \nu_k$ for $k \neq 1$. Under Assumption 1, for any quantile $q \in (0,1)$ and any sequence (b_T, n_T) satisfying*

$$\frac{b_T}{\log(T)} \to +\infty \text{ and } n_T \to +\infty \ ,$$

*the regret of QoMax-ETC with parameters (q, b_T, n_T) is **vanishing in the strong sense**. Furthermore, for polynomial/exponential tails with positive tail gaps this result also holds for $b_T = \Omega(\log T)$.*

Proof. From Theorem 1, there exists constants c_k for $k \geq 2$ such that for T large enough (such that n_T becomes larger than $n_{\nu_1,\nu_k,q}$), it holds that

$$\mathbb{P}(I_T \neq 1) \leq \sum_{k=2}^K \mathbb{P}(\bar{X}^q_{k,n_T,b_T} > \bar{X}^q_{1,n_T,b_T}) \leq \sum_{k=2}^K e^{-c_k b_T}$$

It follows that $\mathbb{P}(I_T \neq 1) = o(T^{-1})$ if $b_T/\log(T) \to \infty$ and we conclude with Proposition 1 and Assumption 1. For polynomial or exponential tails, as the above inequality holds for any value of c_k, $b_T = \Omega(\log T)$ is sufficient to obtain $\mathbb{P}(I_T \neq 1) = o(T^{-1})$. □

Even if Theorem 2 is stated in an asymptotic way, we emphasize that its proof provides a finite-time upper bound on the probability of picking a wrong arm, $\mathbb{P}(I_T \neq 1)$, that is valid provided that T is larger than some (problem-dependent) constant. In particular, T needs to be large enough so that $n_T \geq \max_{k\neq 1} n_{\nu_1,\nu_k,q}$ where $n_{\nu_1,\nu_k,q}$ is the number of samples need in Theorem 1 for the concentration of QoMax. This number is not always large. For example if we have two Pareto distributions with parameters $\lambda_1 = 1.5$ and $\lambda_2 = 2$, $n_T = 3$ is enough. Using our regret decomposition, this result would lead to a finite-time upper bound on the extremal regret for distributions for which a finite-time bound on the exploration cost is available.

For satisfying the theoretical requirements while obtaining good empirical performance, we recommend using $b_T = (\log(T))^2$ and $n_T = \log(T)$ when running the algorithm. All the experiments reported in Section 5 use these values. QoMax-ETC is computationally appealing and has strong asymptotic guarantees. However in practice we found that its performance can vary significantly depending on the choices of b_T and n_T, which should in particular use a reasonable guess for the horizon T. For this reason, in the next section we propose QoMax-SDA, which is still based on QoMax comparisons but is anytime (i.e. independent on T) and requires less parameter tuning.

4 QoMax-SDA

In this section we present QoMax-SDA, an algorithm using a subsampling mechanism based on LB-SDA (Baudry et al., 2021). We detail the key principles of the algorithm and propose a theoretical analysis.

4.1 Algorithm and Implementation

From a high level QoMax-SDA follows the structure of the subsampling duelling algorithms introduced in Baudry et al. (2020). The algorithm operates in successive rounds composed of (1) the selection of a leader, (2) the different duels between the leader and the challengers and (3) a data collection phase. We develop each of those steps in the sequel.

At the beginning of a round r, the learner has access to the history of the different arms denoted $\mathcal{X}_k^r$. For the needs of the QoMax, the collected rewards for arm k are gathered within $b_k(r)$ batches of equal size $n_k(r)$ such that $|\mathcal{X}_k^r| = b_k(r)n_k(r)$. $n_k(r)$ is called the *number of queries* and corresponds to the number of times the arm k has been selected by the learner at the end of round r. The leader at round r, denoted by $\ell(r)$, is the arm that has been queried the most up to round r. The $K-1$ remaining arms are called *challengers*. In case of equality, ties are broken according to any fixed rule (e.g at random). Formally, $\ell(r) = \text{argmax}_{k\leq K}\, n_k(r)$.

Once the leader is selected, $K-1$ duels with the different challengers are performed. We denote $\mathcal{A}_{r+1}$ the set of arms that will be pulled at the end of round r. An arm k is added to $\mathcal{A}_{r+1}$ in two cases (1) if it wins its duel or (2) if its number of queries is too small: $n_k(r) \leq f(r)$ for a fixed function $f(r)$ representing the *sampling obligation*. If no challenger is added to $\mathcal{A}_{r+1}$ the leader is pulled. We now detail the duel procedure that is reported in Algorithm 3. We assume that an infinite stream of rewards is available for each arm, in the form of an array with an infinite number of rows and columns, so that we denote the rewards of arm k by $(X_{k,n,b})_{n\in\mathbb{N},b\in\mathbb{N}}$, where $X_{k,n,b}$ corresponds to the n-th sample of b-th batch from arm k. We further assume that the number of batches available for an arm k depends only on its number of queries $n_k(r)$ so that $b_k(r) = \lceil B(n_k(r))\rceil$ for some function B. The duel is a comparison of the QoMax of the challenger using its entire history and the QoMax of the leader on a subsample of its history.

Algorithm 3 Duel (q-QoMax comparison)

Input: q, arm k, leader ℓ, current history, batch count and batch size: $(\mathcal{X}_m, b_m, n_m)$ for $m \in \{k, \ell\}$

QoMax computation:

1. Compute $I_k = \text{QoMax}(q, b_k, n_k, \mathcal{X}_k)$ (Alg. 1)
2. Collect the **subsample** $\mathcal{Y}_\ell = (X_{\ell,i,j})_{i\in\mathcal{N},j\in\mathcal{B}} \subset \mathcal{X}_\ell$ for $\mathcal{N} = [n_\ell - n_k + 1, n_\ell]$ and $\mathcal{B} = [1, b_k]$.
3. Compute $I_\ell = \text{QoMax}(q, b_k, n_k, \mathcal{Y}_\ell)$ (Alg. 1)

Return: $\text{argmax}_{m\in\{k,\ell\}} I_m$

Our subsampling mechanism is inspired by LB-SDA and works as follows. When comparing the leader $\ell(r)$ with a challenger k: (1) we only consider the rewards collected from the $n_k(r)$ **last** *queries* of arm $\ell(r)$ (as in LB-SDA), and (2) we only keep the $b_k(r)$ **first** batches for $\ell(r)$. This way, the QoMax from the leader and the challenger is computed using the same amount of data. Taking the last queries introduces some diversity in the subsamples encountered when ℓ is often pulled (we refer to Baudry et al. (2021) for details) and using the first batches allows for a reduction of the storage need (see Implementation tricks).

We now detail the data collection procedure that is used by QoMax-SDA and illustrated on Figure 1. If we query arm k with parameters $(\mathcal{X}_k, n_k, b_k, B, \ell)$ at round r, (1) we update existing batches: collect a (n_k+1)-st query for all existing batches $(X_{k,n_k+1,b})_{b\leq b_k}$. (2) Create new batches: *while* $b_k < B(n_k+1)$, collect

the $n_k + 1$ rewards $(X_{k,n,b})_{n \leq n_k+1, b_k < b \leq B(n_k+1)}$.

Combining all those elements gives QoMax-SDA reported in Algorithm 4.

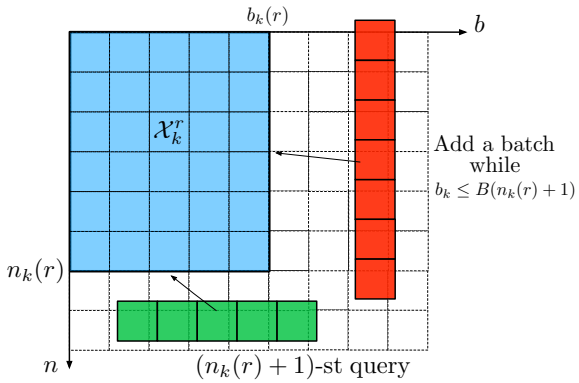

Figure 1: Illustration of the CollectData procedure at round r for a challenger $k \in \mathcal{A}_{r+1}$.

Algorithm 4 QoMax-SDA

Input: K arms, quantile level q
exploration function f, batch function B
Initialization: $r \leftarrow 0$
$\forall k \in \{1, ..., K\}$: $n_k \leftarrow 0$, $b_k \leftarrow 0$, $\mathcal{X}_k \leftarrow$emptyarray2d
for $r \geq 1$ **do**
  $r \leftarrow r + 1$, $\mathcal{A} \leftarrow \{\}$, $\ell \leftarrow \text{leader}(n_k, \mathcal{X}_k)$
  if $r = 1$ **then**
    $\mathcal{A} \leftarrow \{1, \dots, K\}$ (Draw each arm once)
  else
    for $k \neq \ell \in \{1, ..., K\}$ **do**
      if $n_k < f(r)$ *or* $\text{Duel}(k, \ell) = k$ **then**
        $\mathcal{A} \leftarrow \mathcal{A} \cup \{k\}$
      if $|\mathcal{A}| = 0$ **then**
        $\mathcal{A} \leftarrow \{l\}$
  for $k \in \mathcal{A}$ **do**
    $\text{CollectData}(\mathcal{X}_k, n_k, b_k, B, \ell)$, update $\mathcal{X}_k$, b_k
    $n_k \leftarrow n_k + 1$

Implementation tricks Our algorithm can enjoy a significant reduction of storage with two different tricks. (1) An efficient storing of the maxima: for arm k in the batch b every time a new sample x is collected, all stored values smaller than x (if any) are deleted. The new sample x and the round where x was received are then stored. (2) An efficient CollectData procedure. We could use the same procedure for all the arms and obtain a number of batch for the leader that scales as n^γ. If the algorithm ends up pulling an arm most of the time (which is expected), this will create new batches for the leader that are never used in the duels because with our subsampling mechanism, only the first b_k batches are used when the leader competes with arm k. Instead, the CollectData procedure is only applied to the challengers (see Algorithm 6) and a batch is added to the leader only when it has to match the number of batches of the second most pulled arm. Those tricks are detailed in Appendix D.1.

Note that a sampling obligation, through the exploration function f (independent on T), is necessary under general assumptions as in all existing algorithms.

4.2 Extreme Regret Analysis

We now provide an analysis of QoMax-SDA under the same assumption as before: $\nu_1 \succ \nu_k$ for all $k \neq 1$. Let $N_k(t)$ denote the number of pulls of arm k at time t. We start with a generic regret decomposition.

Proposition 2 (Regret decomposition with a low probability event)**.** *Define the event*

$$\xi_T := \{N_1(T) \leq T - KM_T\} ,$$

where $(M_T)_{T \in \mathbb{N}}$ is a fixed sequence. Then, for $T \geq KM_T$, for any constant $x_T \in \mathbb{R}$, it holds that,

$$\mathcal{R}_T^\pi \leq \underbrace{\mathbb{E}\left[\max_{t \leq T} X_{1,t}\right] - \mathbb{E}\left[\max_{t \leq T - KM_T} X_{1,t}\right]}_{\textit{Exploration cost}} + \underbrace{x_T \mathbb{P}(\xi_T) + \mathbb{E}\left[\max_{t \leq T} X_{1,t} \mathbb{1}\left(\max_{t \leq T} X_{1,t} \geq x_T\right)\right]}_{\textit{Cost incurred by } \xi_T} .$$

The proof of this result follows the analysis from Carpentier and Valko (2014) and is given in Appendix C, which contains the proofs of all results from this section. The "cost incurred by ξ_T" features two terms. Interestingly, only the first term depends on the algorithm. We upper bound it below.

Lemma 3 (Upper bound on $\mathbb{P}(\xi_T)$)**.** *For any $q \in (0, 1)$, any M_T and any $\gamma > 0$, under QoMax-SDA with parameters $B(n) = n^\gamma$ and $f(r) = (\log r)^{\frac{1}{\gamma}}$,*

$$\mathbb{P}(\xi_T) = \mathcal{O}\left((\log T)^{\frac{1}{\gamma}} M_T^{-\frac{1}{1+\gamma}}\right) .$$

Moreover, for all $k \neq 1$, $\mathbb{E}[n_k(T)] = \mathcal{O}((\log T)^{1/\gamma})$.

Sketch of proof. We first prove that $\mathbb{P}(\xi_T)$ is upper bounded by $\sum_{k=2}^K \mathbb{P}(N_k(T) \geq M_T)$. Using that $N_k(T) = b_k(T) \times n_k(T) = n_k(T)^{1+\gamma}$ and Markov inequality we obtain

$$\mathbb{P}(\xi_T) \leq M_T^{-\frac{1}{1+\gamma}} \sum_{k=2}^K \mathbb{E}[n_k(T)] .$$

It remains to study the expected number of queries of sub-optimal arms $k \geq 2$. This can be done following the outline of Baudry et al. (2021) and using the deviation inequalities from Theorem 1. □

The second term in the "cost incurred by ξ_T" only depends on the distribution of the optimal arm and can be further upper bounded assuming exponential and polynomial tails, leading to the following result.

Theorem 3 (Upper bound on the regret of QoMax-SDA)**.** *For any quantile q, any $\gamma > 0$, defining the parameters of QoMax-SDA as $B(n) = n^\gamma$ and $f(r) = (\log r)^{\frac{1}{\gamma}}$. The regret of QoMax-SDA is* ***(1)*** *vanishing in the strong sense for exponential tails* ***(2)*** *vanishing in the weak sense for polynomial tails.*

Sketch of proof. For parametric tails, we can calculate the growth rate of $\mathbb{E}[\max_{t\leq T} X_{1,t}]$ with respect to T. This permits to tune the values of M_T and x_T to properly balance the terms in the regret decomposition. The difference in the convergence for **(1)** and **(2)** comes from the fact that the exploration cost scales logarithmically with the time horizon when using exponential tails, whereas the dependency is polynomial with polynomial tails. □

We note that there is no hope to upper bound the last term in our current regret decomposition assuming only that arm 1 dominates the others, so we could not establish vanishing regret for QoMax-SDA under this assumption. As can be seen in the proof of Proposition 2, the "cost incurred by ξ_T" is actually an upper bound on $\mathbb{E}\left[\mathbb{1}(\xi_T)\max_{t\leq T} X_{1,t}\right]$. If this term were upper bounded by $\mathbb{P}(\xi_T)\mathbb{E}\left[\max_{t\leq T} X_{1,t}\right]$[1], we would get a regret decomposition closer to that in Proposition 2, leading to a strongly vanishing regret for QoMax-SDA using Lemma 3. Even if we were not able to prove this, we note that (1) QoMax-SDA achieves state-of-the-art performance for exponential and polynomial tails (2) Lemma 3 provides a strong indicator of the good performance of QoMax-SDA under more general assumptions, as it shows that the algorithm queries each sub-optimal arm $\mathcal{O}((\log T)^{\frac{1}{\gamma}})$ times.

We now turn our attention to the practical benefits of using our QoMax-based algorithms.

5 PRACTICAL PERFORMANCE

In all of our experiments, we compare QoMax-SDA and QoMax-ETC with ThresholdAscent (Streeter and Smith, 2006b), ExtremeHunter (Carpentier and Valko, 2014), ExtremeETC (Achab et al., 2017) and MaxMedian (Bhatt et al., 2021). We use the parameters suggested in the original papers (see Appendix D for details and remarks on the tuning). Namely, $b = 1$ for ExtremeHunter/ETC, $s = 100, \delta = 0.1$ for ThresholdAscent, $\varepsilon_t = (t+1)^{-1}$ for MaxMedian. For QoMax-ETC, we use $b_T = (\log T)^2$ batches of $n_T = \log T$ samples. This matches the size of the exploration phase of ExtremeETC and allows for a fair comparison. For QoMax-SDA, we choose $\gamma = 2/3$, which seems to work well across all examples. All the results presented in this section are obtained with these values.

5.1 Time and Memory Complexity

We summarize in Table 1 the storage and computational time required by the different adaptive and ETC algorithms that we consider, with the aforementioned parameters. The smallest values in each category are colored in blue. We do not include ThresholdAscent in the table because the comparison is unfair, as it uses a fixed number of data but is not theoretically grounded. We refer the reader to Bhatt et al. (2021) for the complexities of the baselines, and we give a few insights on how we obtained the results for QoMax algorithms (details can be found in Appendix D.2).

For QoMax-ETC, the memory needed is Kb_T and the time complexity is in $\mathcal{O}(\max(n_T\ \ b_T, b_T \log b_T))$ due to the collection phase and the quantile computation. Plugging the values of b_T and n_T gives the result. The time complexity of QoMax-SDA is in $\mathcal{O}(KT \log T)$ as its main cost consists in sorting data online, just like MaxMedian. The storage of QoMax-SDA is obtained thanks to the two tricks: one allows to keep $\mathcal{O}(\log T)$ batches, the other $\mathcal{O}(\log T)$ samples per batch *for the leader*. On the contrary, the complexity for the challengers remains in $\mathcal{O}(\log T \log\log T)$, therefore the dependency in K only appears as a second order term.

Table 1: Average time and storage complexities of Extreme Bandit algorithms for a time horizon T.

Algorithm	Memory	Time
Extreme Hunter	T	$\mathcal{O}(T^2)$
MaxMedian	T	$\mathcal{O}(KT \log T)$
QoMax-SDA	$\mathcal{O}((\log T)^2)$	$\mathcal{O}(KT \log T)$
Extreme ETC	$\mathcal{O}\left(K(\log T)^3\right)$	$\mathcal{O}\left(K(\log T)^6\right)$
QoMax-ETC	$\mathcal{O}(K(\log T)^2)$	$\mathcal{O}(K(\log T)^3)$

QoMax-SDA offers an exponential reduction of the storage cost compared to ExtremeHunter and MaxMedian, while being as computationally efficient as MaxMedian. On the other hand, choosing the same length for the exploration phase of the two ETC leads to a significantly smaller time complexity for QoMax-ETC. Hence, both QoMax-SDA and QoMax-ETC present a substantial improvement over their counterparts.

[1]This is intuitively true as under ξ_T, arm 1 underperforms, hence $\mathbb{1}(\xi_T)$ and $X^+_{1,T}$ are expected to be negatively correlated

5.2 Empirical Performance

We compare the empirical performance of the QoMax algorithms with the different competitors on synthetic data. We reproduced 6 experiments from previous works[2]: all experiments from Bhatt et al. (2021) (Experiments 1-4 for us), and the experiments 1 and 2 from Carpentier and Valko (2014) (5-6 here). We also implement new experiments with other families of distributions to highlight the generality of our approach. Due to space limitation, we present in this section (1) our methodology for evaluating Extreme Bandits algorithms, and (2) the results for the experiment 1 of Bhatt et al. (2021), as it illustrates well our findings across all the settings we tested. We analyze the results for the other experiments in Appendix D.

Empirical evaluation We consider 4 performance criteria: **(I)** an *empirical evaluation of the extreme regret*, **(II)** the *fraction of pulls of the optimal arm*, **(III)** the *empirical distribution of the number of pulls of the optimal arm* and **(IV)** the *empirical distribution of the maximal reward*, estimated over $N = 10^4$ independent trajectories for different values of the horizon T. Most works report only **(I)**, and **(II)** was first proposed by Bhatt et al. (2021). Our analysis shows that the extreme regret of a strategy is closely related to its capacity to sample the optimal arm $T - o(T)$ times, so we think that **(II)** is indeed a good performance indicator. Criterion **(III)** completes it by displaying the following quantiles of the empirical distribution of best arm pulls: $q \in [1\%, 10\%, 25\%, 50\%, 75\%, 90\%, 99\%]$. Regarding **(I)**, we note that estimating the expectation $\mathbb{E}[\max_{t\leq T} X_{I_t,t}]$ featured in the extreme regret is very hard, and that approximations of $\mathbb{E}[X^+_{1,T}] \coloneqq \mathbb{E}[\max_{t\leq T} X_{1,t}]$ are known only for a few families. Standard Monte-Carlo estimators will have a very large variance due to the heavy tails of the distributions (see illustrations in Appendix D). Hence, we propose the following estimation strategy *when a tight approximation of* $\mathbb{E}[X^+_{1,T}]$ *is known.* We first find $\tilde{q} = \tilde{q}_{\nu_1,T}$ such that $\mathbb{E}[X^+_{1,T}]$ is equal to the quantile of order $\tilde{q}_{\nu_1,T}$ of $\nu^+_{1,T}$. We then compute the empirical quantile of order $\tilde{q}$ of the collected rewards, denoted by $\widehat{X}_T(q)$, as an estimator of their expected maximum. This allows to compute what we call Proxy Empirical Regret (PER), $\mathcal{R}_T^{\text{proxy}} = \frac{\mathbb{E}[X^+_{1,T}] - \widehat{X}_T(q)}{\mathbb{E}[X^+_{1,T}]}$, where the normalization facilitates the check of a *weakly vanishing* regret. We are able to compute **(I)** for experiments 1-6. When **(I)** is not available we recommend looking at **(IV)** with the same quantiles as for **(III)**.

Experiment 1 We consider $K = 5$ Pareto distributions with parameters $[1.1, 1.3, 1.9, 2.1, 2.3]$. We chose this experiment because it enters in the theoretical guarantees of most baselines. The QoMax-based algorithms outperform their competitors in this problem, both in terms of **(I)** and **(II)** (see Figure 2). QoMax-SDA learns faster, but at horizon $T = 5 \times 10^4$ the ETC are close. The quantile $q = 0.5$ performs (very) slightly better than $q = 0.9$. Strikingly, QoMax algorithms surpass the two baselines designed for this parametric setting (ExtremeHunter, ExtremeETC). We also observe that the performance of ThresholdAscent and MaxMedian stops improving early, even if MaxMedian is competitive for $T \leq 10^4$. To understand this phenomenon, we look at **(III)** (Table 3 in Appendix D). Surprisingly, for at least 25% of the trajectories MaxMedian ended up playing the optimal arm less than 35 times over 5×10^4 pulls[3]. On the other hand, QoMax-SDA ($q = 0.5$) selects the optimal arm at least 2×10^4 times for 99% of them. It also obtains much higher statistics on the empirical distribution of the maxima (see Table 4 in Appendix D, criterion **(IV)**).

Other Experiments The benefits of QoMax are also clear from experiments 2 to 5: we verify that they work well for Exponential (experiment 3), Gaussian tails (experiment 4), as well as for other Pareto examples (experiments 2 and 5) including one where the tail gap is 0 (experiment 2). The impact of the number of arms is discussed, showing that for reasonable time horizons QoMax-SDA should be preferred over QoMax-ETC (experiment 4). Experiment 6 allows to discuss the limits of QoMax in a difficult scenario, in which the 2nd-order Pareto assumption allows ExtremeHunter and ExtremeETC to outperform all other algorithms. In this example, setting $q = 0.9$ has a benefit as well as enforcing the sample obligation. Finally, in additional experiments we consider different families of heavy-tail distributions stressing out the generality of the dominance assumption under which QoMax algorithms are efficient.

Conclusion Overall, QoMax-based algorithms seem to be solid choices for the practitioner, as demonstrated in a variety of examples. Their strong theoretical guarantees and implementation tricks reducing the time and space complexities make them an efficient solution for the Extreme Bandits problem.

[2]Our code is available here

[3]Furthermore, we discuss in Appendix C a potential issue in the analysis of MaxMedian

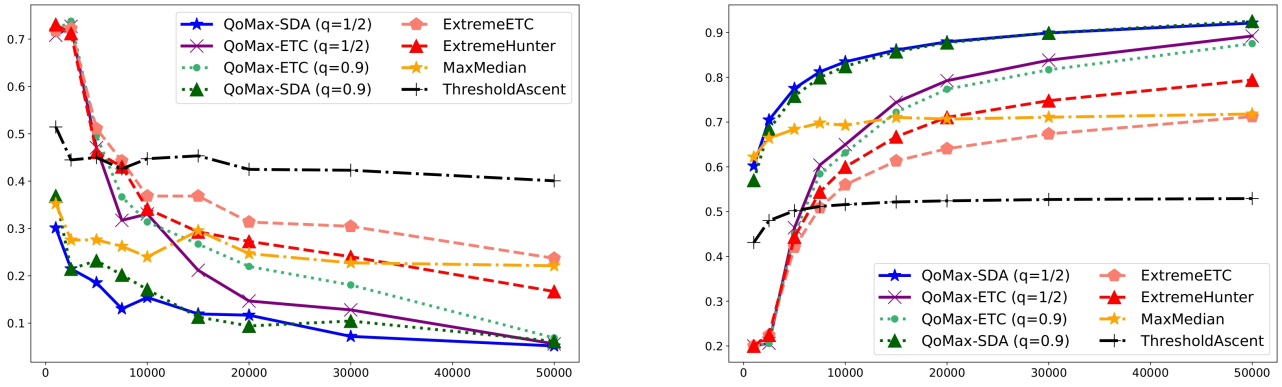

Figure 2: Proxy Empirical Regret **(I)** and Percentage of best arm pulls **(II)** averaged over 10^4 independent trajectories for $T \in \{10^3, 2.5\times10^3, 5\times10^3, 7.5\times10^3, 9\times10, 10^4, 1.5\times10^4, 2\times10^4, 3\times10^4, 5\times10^4\}$.

Acknowledgements

The PhD of Dorian Baudry is funded by a CNRS80 grant. This work has been supported by the French Ministry of Higher Education and Research, Inria, Scool, and the French Agence Nationale de la Recherche (ANR) under grant ANR-19-CE23-0026-04 (BOLD project).

Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

The authors want to thank Mastane Achab, Alexandra Carpentier and Michal Valko for carefully answering our questions regarding the ExtremeHunter and ExtremeETC algorithms, and helping us for their implementation.

References

M. Achab, S. Clémençon, A. Garivier, A. Sabourin, and C. Vernade. Max k-armed bandit: On the extremehunter algorithm and beyond. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 389–404. Springer, 2017.

N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and system sciences*, 58(1): 137–147, 1999.

D. Baudry, E. Kaufmann, and O.-A. Maillard. Subsampling for efficient non-parametric bandit exploration. *Advances in Neural Information Processing Systems*, 33, 2020.

D. Baudry, Y. Russac, and O. Cappé. On Limited-Memory Subsampling Strategies for Bandits. In *ICML 2021- International Conference on Machine Learning*, Vienna / Virtual, Austria, July 2021.

S. Bhatt, P. Li, and G. Samorodnitsky. Extreme bandits using robust statistics. *arXiv preprint arXiv:2109.04433*, 2021.

S. Bubeck, N. Cesa-Bianchi, and G. Lugosi. Bandits with heavy tail. *IEEE Transactions on Information Theory*, 59(11):7711–7717, 2013.

A. Carpentier and M. Valko. Extreme bandits. In *Neural Information Processing Systems*, Montréal, Canada, Dec. 2014.

H. P. Chan. The multi-armed bandit problem: An efficient nonparametric solution. *The Annals of Statistics*, 48(1):346–373, 2020.

V. A. Cicirello and S. F. Smith. The max k-armed bandit: A new model of exploration applied to search heuristic selection. In *The Proceedings of the Twentieth National Conference on Artificial Intelligence*, volume 3, pages 1355–1361, 2005.

M. Gilli et al. An application of extreme value theory for measuring financial risk. *Computational Economics*, 27(2):207–228, 2006.

D. B. Neill and G. F. Cooper. A multivariate bayesian scan statistic for early event detection and characterization. *Machine learning*, 79(3):261–282, 2010.

R. Nishihara, D. Lopez-Paz, and L. Bottou. No regret bound for extreme bandits. In *Artificial Intelligence and Statistics*, pages 259–267. PMLR, 2016.

B. Skiera, J. Eckert, and O. Hinz. An analysis of the importance of the long tail in search engine marketing. *Electronic Commerce Research and Applications*, 9(6):488–494, 2010.

M. J. Streeter and S. F. Smith. An asymptotically optimal algorithm for the max k-armed bandit problem. In *AAAI*, pages 135–142, 2006a.

M. J. Streeter and S. F. Smith. A simple distribution-free approach to the max k-armed bandit problem. In *International Conference on Principles and Practice of Constraint Programming*, pages 560–574. Springer, 2006b.

Supplementary Material: Efficient Algorithms for Extreme Bandits

A PROPERTIES OF MAXIMA

We first recall the notation from Section 2. We consider i.i.d. samples $(X_{1,i})$ and $(X_{2,i})$ from two distributions ν_1 and ν_2 and denote by $X_{1,i}^+$ and $X_{2,i}^+$ their maxima. Our goal is to upper bound

$$\max\left\{\mathbb{P}\left(X_{1,n}^+ \leq x_n\right), \mathbb{P}\left(X_{2,n}^+ \geq x_n\right)\right\}$$

for a well chosen sequence (x_n) under exponential and polynomial tails (Lemma 1) and under the weaker assumption that $\nu_1 \succ \nu_2$ (Lemma 2). In both cases, we start by writing

$$\mathbb{P}\left(X_{1,n}^+ \leq x_n\right) = (1 - G_1(x_n))^n \leq \exp(-nG_1(x_n)) \tag{4}$$

$$\mathbb{P}\left(X_{2,n}^+ \geq x_n\right) \leq \sum_{i=1}^{n} \mathbb{P}(X_{2,i} \geq x_n) = nG_2(x_n), \tag{5}$$

where G_1 and G_2 are the survival functions of ν_1 and ν_2 respectively.

A.1 Proof of Lemma 1

Lemma 1 (Comparison of Maxima)**.** *Assume that both ν_1 and ν_2 have either polynomial or exponential tails, with respective second parameter λ_1 and λ_2, with $\lambda_1 < \lambda_2$ (so that $\nu_1 \succ \nu_2$). Define $\delta = \frac{\lambda_2}{\lambda_1} - 1 > 0$, then there exists a sequence (x_n) such that*

$$\max\{\mathbb{P}(X_{1,n}^+ \leq x_n), \mathbb{P}(X_{2,n}^+ \geq x_n)\} = \mathcal{O}\left(\frac{(\log n)^{\delta+1}}{n^\delta}\right).$$

Proof. The key of the proof is to consider x_n "slightly" below $G_1^{-1}(1/n)$. Consider the exponential tails first, for which $G_1(x) \sim C_1 \exp(-\lambda_1 x)$ and $G_2(x) \sim C_2 \exp(-\lambda_2 x)$, for some (C_1, λ_1) and (C_2, λ) with $\lambda_1 < \lambda_2$. Hence, for any $\varepsilon > 0$ it holds that for x large enough, $G_1(x) \geq (1-\varepsilon)C_1 \exp(-\lambda_1 x)$ and $G_2(x) \leq C_2(1+\varepsilon)\exp(-\lambda_2 x)$. So, we prove without loss of generality the result by continuing the proof as if the survival functions were exactly equal to their equivalents, as we don't assume anything on C_1 and C_2.

We let $\delta = \frac{\lambda_2}{\lambda_1} - 1$ and choose

$$x_n = \frac{1}{\lambda_1}\left(\log n + \log(C_1) - \log(\delta \log n)\right) .$$

We now simply compute $G_1(x_n)$ and $G_2(x_n)$. First,

$$\begin{aligned} G_1(x_n) &= C_1 \exp(-(\log n + \log C_1 - \log(\delta \log n)) \\ &= \frac{\delta(\log n)}{n} . \end{aligned}$$

Then,

$$\begin{aligned} G_2(x_n) &= C_2 \exp\left(-\frac{\lambda_2}{\lambda_1}\left(\log n + \log C_1 - \log(\delta \log n)\right)\right) \\ &= \frac{1}{n^{\frac{\lambda_2}{\lambda_1}}} \times (\delta \log n)^{\frac{\lambda_2}{\lambda_1}} \times \frac{C_2}{C_1^{\frac{\lambda_2}{\lambda_1}}} \end{aligned}$$

So finally, using Equation (4) and (5) we obtain

$$\mathbb{P}\left(Y_n^+ \geq x_n\right) = \mathcal{O}\left(\frac{(\log n)^{\delta+1}}{n^\delta}\right) \quad \text{and} \quad \mathbb{P}\left(X_n^+ \leq x_n\right) \leq \frac{1}{n^\delta} \,,$$

which gives the result.

Now we consider polynomial tails, for which $G_1(x) = C_1 x^{-\lambda_1}$ and $G_2(x) = C_2 x^{-\lambda_2}$ for x large enough. This time we define the sequence

$$x_n = (C_1 n)^{\frac{1}{\lambda_1}} \times (\delta \log n)^{-\frac{1}{\lambda_1}} \,,$$

with $\delta = \frac{\lambda_2}{\lambda_1} - 1$, as above. We obtain $\exp(-nG_1(x_n)) = n^{-\delta}$, and $nG_2(x_n) = \mathcal{O}\left(\frac{(\log n)^{\delta+1}}{n^\delta}\right)$, giving the result. □

A.2 Proof of Lemma 2

Lemma 2. *Assume that $\nu_1 \succ \nu_2$. Then, for any $q \in (0,1)$ there exists $n_{\nu_1,\nu_2,q} \in \mathbb{N}$, a sequence x_n and some $\varepsilon > 0$ such that for all $n \geq n_{\nu_1,\nu_2,q}$ and b large enough,*

$$\mathbb{P}(X_{1,n}^+ \leq x_n) \leq q - \varepsilon \,, \quad \textit{and} \quad \mathbb{P}(X_{2,n}^+ \leq x_n) \geq q + \varepsilon \,.$$

Proof. Let $q \in (0,1)$. We define the sequence (x_n) by

$$G_1(x_n) = 1 - q^{\frac{1}{n}},$$

so that $\mathbb{P}(X_{1,n}^+ \leq x_n) = q$.

As $\nu_1 \succ \nu_2$, there exists a constant $C > 1$ such that $G_1(x) \geq CG_2(x)$ for x large enough. Hence, as $x_n \to +\infty$ it holds that $G_1(x_n) > CG_2(x_n)$ for n large enough.

For such large enough n we have

$$\begin{aligned}
\mathbb{P}(X_{2,n}^+ \leq x_n) &= (1 - G_2(x_n))^n \\
&\geq (1 - \frac{1}{C} G_1(x_n))^n \\
&= \left(1 - \frac{1}{C}\left(1 - q^{\frac{1}{n}}\right)\right)^n .
\end{aligned}$$

Now we can consider the asymptotic behavior of this quantity, using first that $1 - q^{\frac{1}{n}} \sim \frac{-\log q}{n}$, and deducing that

$$\lim_{n\to\infty} \left(1 - \frac{1}{C}\left(1 - q^{\frac{1}{n}}\right)\right)^n = q^{\frac{1}{C}} \,.$$

Hence, for any $\varepsilon_0 > 0$ when n is large enough we have for this specific choice of x_n

$$\mathbb{P}(X_{n,1}^+ \leq x_n) = q \quad \text{and} \quad \mathbb{P}(X_{2,n}^+ \geq x_n) < 1 - q^{\frac{1}{C}} + \varepsilon_0 \,.$$

It is clear from this point that if we change a bit x_n in order to have $\mathbb{P}(X_{n,1}^+ \leq x_n) \leq q - \varepsilon$, for some $\varepsilon > 0$, then $\mathbb{P}(X_{2,n}^+ \geq x_n) \leq 1 - (q-\varepsilon)^{\frac{1}{C}} + \varepsilon$ holds for n large enough. Taking ε small enough to obtain $1 - (q-\varepsilon)^{\frac{1}{C}} + \varepsilon < 1 - q - \varepsilon$ concludes the proof. □

A.3 Lower Bound

A natural question is whether the rate obtained in Lemma 1 can be improved, and if it is really impossible to achieve an exponentially decreasing probability as for the comparison of empirical means. We show that this is not the case even under semi-parametric assumptions with the following result.

Lemma 4 (Lower bound)**.** *Assume that both ν_1 and ν_2 have either polynomial or exponential tails, with respective second parameter λ_1 and λ_2, with $\lambda_1 < \lambda_2$ (so that $\nu_1 \succ \nu_2$).*

$$\mathbb{P}\left(X_{1,n}^+ \leq X_{2,n}^+\right) = \Omega\left(n^{-\frac{\lambda_2}{\lambda_1}}\right) \,.$$

Proof. Letting f_k and F_k be the pdf and cdf of the distribution ν_k for $k \in \{1,2\}$. We lower bound the probability of interest as follows:

$$\begin{aligned}\mathbb{P}(X_{1,n}^+ \leq X_{2,n}^+) &\geq \mathbb{P}(X_{2,1} \geq \max_{1\leq i\leq n} X_{1,i}) \\ &= \mathbb{E}_{X\sim\nu_2}[F_1(X)^n] = \int_{\mathbb{R}} f_2(x)F_1(x)^n dx \\ &\geq \int_{m_n}^{+\infty} f_2(x)F_1(x)^n dx \geq F_1(m_n)^n G_2(m_n)\ ,\end{aligned}$$

for any choice of m_n. If we choose $m_n = F_1^{-1}(1-\frac{1}{n})$, we have for exponential tails $m_n = \frac{1}{\lambda_1}(\log C_1 + \log n)$. If we plug this into G_2 we obtain a lower bound in $e^{-1}\frac{C_2}{C_1^{\lambda_2/\lambda_1}}\frac{1}{n^{\frac{\lambda_2}{\lambda_1}}}$. The same can be done for polynomial tails. □

A.4 Maxima of (semi)-parametric distributions

We first introduce a few notation to ease the presentation. In this section, we let $(X_i)_{i\in\mathbb{N}}$ be an i.i.d. sequence from the distribution ν_1 whose survival function is denoted by G_1. For any integers n, m with $n < m$, we let $X_{n:m}^+ = \max_{i\in\{n,\dots,m\}} X_i$ and use the shorthand $X_n^+ = X_{1:n}^+$.

We first recall known results about the rate of growth of the expected maximum for distributions that have exponential or polynomial tails.

Proposition 3. *If ν_1 has an exponential tail with parameters C_1 and λ_1, it holds that*

$$\mathbb{E}\left[X_T^+\right] \underset{T\to\infty}{\sim} \frac{1}{\lambda_1}\log(T)$$

If ν_1 has a polynomial tail with parameters C_1 and $\lambda_1 > 1$, it holds that

$$\mathbb{E}\left[X_T^+\right] \underset{T\to\infty}{\sim} T^{\frac{1}{\lambda_1}} C_1^{\frac{1}{\lambda_1}} \Gamma\left(1-\frac{1}{\lambda_1}\right)$$

Proof. For exponential tails, we refer the reader to Appendix A.1 of Bhatt et al. (2021). For polynomial tails, we can use Theorem 1 of Carpentier and Valko (2014) which applies to second-order Pareto distributions, and in particular Pareto distributions, for which $G(x) = \frac{C}{x^\lambda}$ for x large enough, with exact equality. To handle our semi-parametric assumption, we first note that for all B,

$$\begin{aligned}\mathbb{E}\left[X_T^+\right] &= \mathbb{E}\left[X_T^+ \mathbb{1}\left(X_T^+ \leq M\right)\right] + \mathbb{E}\left[X_T^+ \mathbb{1}\left(X_T^+ > M\right)\right] \\ &= \mathbb{E}\left[X_T^+ \mathbb{1}\left(X_T^+ \leq M\right)\right] + \int_M^\infty \left(1-(1-G_1(x))^T\right) dx\end{aligned}$$

The first terms tends to zero when T goes to infinity for any distribution that has an unbounded support, so if $G_{C,\lambda}(x) = \frac{C}{x^\lambda}$ is the survival function of an exact Pareto distribution, it follows that for all $M > 0$

$$\int_M^\infty \left(1-(1-G_{C,\lambda}(x))^T\right) dx \sim T^{\frac{1}{\lambda_1}} C_1^{\frac{1}{\lambda}} \Gamma\left(1-\frac{1}{\lambda}\right).$$

Now assume that $G_1(x) \sim C_1 x^{-\lambda_1}$ when x tends to infinity. For all $\varepsilon > 0$ there exists $M > 0$ such that for $x > M$,

$$(1-\varepsilon)\frac{C}{x^\lambda} \leq G_1(x) \leq (1+\varepsilon)\frac{C}{x^\lambda}$$

and

$$\begin{aligned}\mathbb{E}\left[X_T^+\right] &\leq \mathbb{E}\left[X_T^+ \mathbb{1}\left(X_T^+ \leq M\right)\right] + \int_M^\infty \left(1-(1-G_{(1+\varepsilon)C_1,\lambda_1}(x))^T\right) dx \sim T^{\frac{1}{\lambda_1}}((1+\varepsilon)C_1)^{\frac{1}{\lambda_1}}\Gamma\left(1-\frac{1}{\lambda_1}\right) \\ \mathbb{E}\left[X_T^+\right] &\geq \mathbb{E}\left[X_T^+ \mathbb{1}\left(X_T^+ \leq M\right)\right] + \int_M^\infty \left(1-(1-G_{(1-\varepsilon)C_1,\lambda_1}(x))^T\right) dx \sim T^{\frac{1}{\lambda_1}}((1-\varepsilon)C_1)^{\frac{1}{\lambda_1}}\Gamma\left(1-\frac{1}{\lambda_1}\right),\end{aligned}$$

which permits to conclude the proof. □

We now recall Assumption 1, under which we analyse QoMax-ETC and QoMax-SDA.

Assumption 1. $\mathbb{E}\left[\max_{t\leq T} X_{1,t}\right] = o(T)$, *and for any* $\gamma < 1$ *if* $N_T = o(T^\gamma)$ *then*

$$\mathbb{E}\left[\max_{t\leq T} X_{1,t}\right] - \mathbb{E}\left[\max_{t\leq T-KN_T} X_{1,t}\right] \xrightarrow[T\to+\infty]{} 0\,.$$

In order find a sufficient condition for Assumption 1 to be satisfied, for any constant $U > 0$ we write

$$\begin{aligned}
\mathbb{E}[X_T^+] - \mathbb{E}[X_{T-N_T}^+] &= \mathbb{E}\left[X_{T-N_T+1:T}^+ \mathbb{1}\left(X_T^+ = X_{T-N_T+1:T}^+\right)\right] \\
&\leq \mathbb{E}\left[X_{T-N_T+1:T}^+ \mathbb{1}\left(X_T^+ = X_{T-N_T+1:T}^+\right) \mathbb{1}\left(X_{T-N_T+1:T}^+ \leq B\right)\right] \\
&\quad + \mathbb{E}\left[X_{T-N_T+1:T}^+ \mathbb{1}\left(X_{T-N_T+1:T}^+ > B\right)\right] \\
&\leq B\mathbb{P}\left(X_T^+ = X_{T-N_T+1:T}^+\right) + \int_B^\infty \mathbb{P}\left(X_{T-N_T+1:T}^+ > x\right) dx \\
&= B\frac{N_T}{T} + \int_B^\infty \mathbb{P}\left(X_{N_T}^+ > x\right) dx \\
&\leq B\frac{N_T}{T} + N_T \int_B^\infty \mathbb{P}(X_1 > x) dx \\
&\leq N_T\left(\frac{B}{T} + \int_B^\infty G_1(x) dx\right).
\end{aligned}$$

where we have used the fact that that maximum has the same probability to be attained in each batch of size N_T and the union bound $\mathbb{P}\left(X_{N_T}^+ > x\right) \leq \sum_{i=1}^{N_T} \mathbb{P}(X_i > x)$.

To prove that Assumption 1 is satisfied for exponential and polynomial tails, in each case we exhibit a value of B such that the resulting upper bound tends to 0.

Exponential tails In that case $G_1(x) = O\left(C_1 e^{-\lambda_1 x}\right)$ and there exists a constant $C > 0$ such that

$$\begin{aligned}
\mathbb{E}[X_T^+] - \mathbb{E}[X_{T-N_T}^+] &\leq N_T\left(\frac{B}{T} + C\int_B^\infty e^{-\lambda_1 x} dx\right) \\
&= N_T\left(\frac{B}{T} + \frac{C}{\lambda_1} e^{-\lambda_1 B}\right).
\end{aligned}$$

If *there exists* $\gamma \in (0,1)$ such that $N_T = o(T^\gamma)$, choosing $B = \frac{\log(T)}{\lambda_1}$ yields $\lim_{T\to\infty} \mathbb{E}[X_T^+] - \mathbb{E}[X_{T-N_T}^+] = 0$.

Polynomial tails In that case $G_1(x) = O\left(C_1 x^{-\lambda_1}\right)$ for $\lambda_1 > 1$ and there exists a constant $C > 0$ such that

$$\begin{aligned}
\mathbb{E}[X_T^+] - \mathbb{E}[X_{T-N_T}^+] &\leq N_T\left(\frac{B}{T} + C\int_B^\infty \frac{1}{x^{\lambda_1}} dx\right) \\
&= N_T\left(\frac{B}{T} + \frac{C}{\lambda_1} B^{1-\lambda_1}\right)
\end{aligned}$$

Choosing $B = T^{1/\lambda_1}$ yields

$$\mathbb{E}[X_T^+] - \mathbb{E}[X_{T-N_T}^+] \leq \left(1 + \frac{C}{\lambda_1}\right) \frac{N_T}{T^{1-\frac{1}{\lambda_1}}}$$

If *for all* $\gamma \in (0,1)$, $N_T = o(T^\gamma)$ then in particular $N_T = o(T^{1-\frac{1}{\lambda_1}})$ and $\lim_{T\to\infty} \mathbb{E}[X_T^+] - \mathbb{E}[X_{T-N_T}^+] = 0$.

B PROOFS OF SECTION 3 (ETC)

Proposition 1 (Regret of QoMax-ETC). *Let π be an ETC policy sampling $N_T = n_T \times b_T$ times each arm during the exploration phase. If $T \geq KN_T$,*

$$\mathcal{R}_T^\pi \leq \underbrace{\mathbb{E}\left[\max_{t\leq T} X_{1,t}\right] - \mathbb{E}\left[\max_{t\leq T-KN_T} X_{1,t}\right]}_{\textit{Exploration cost}} + \underbrace{\mathbb{P}(I_T \neq 1)\mathbb{E}\left[\max_{t\leq T} X_{1,t}\right]}_{\textit{Cost of picking a wrong arm}} .$$

Proof. We recall that $N_T = b_T \times n_T$ is the number of pulls of each arm during the exploration phase of the ETC algorithm (see Algorithm 2) and that $X_{k,t}$ corresponds to the observation of arm k at time t (if any). The ETC simplifies a lot the study of the extremal regret, as we can separate the explore and commit phase in the analysis. First, an exact decomposition of the expected value of the policy is

$$\mathbb{E}\left[\max_{t\leq T} X_{I_t,t}\right] = \mathbb{E}\left[\max\left\{\max_k \max_{t\leq KN_T} X_{k,t}, \max_{t=[KN_T+1,T]} X_{I_T,t}\right\}\right] .$$

We obtain the lower bound by simply ignoring the exploration phase.

$$\begin{aligned}
\mathbb{E}\left[\max_{t\leq T} X_{I_t,t}\right] &\geq \mathbb{E}\left[\max_{t=[KN_T+1,T]} X_{I_t,t}\right] \\
&= \mathbb{E}\left[\max_{t=[KN_T+1,T]} X_{I_T,t}\right] \\
&= \mathbb{E}\left[\max_{t=[KN_T+1,T]} X_{I_T,t} \sum_{k=1}^K \mathbb{1}(I_T = k)\right] \\
&= \sum_{k=1}^K \mathbb{E}\left[\max_{t=[KN_T+1,T]} X_{I_T,t}\mathbb{1}(I_T = k)\right] \\
&= \sum_{k=1}^K \mathbb{P}(I_T = k)\mathbb{E}\left[\max_{t=[KN_T+1,T]} X_{k,t}\right] \\
&\geq \mathbb{P}(I_T = 1)\mathbb{E}\left[\max_{t=[1,T-KN_T]} X_{1,t}\right] \\
&= (1 - \mathbb{P}(I_T \neq 1))\mathbb{E}\left[\max_{t=[1,T-KN_T]} X_{1,t}\right] \\
&\geq \mathbb{E}\left[\max_{t\leq T-KN_T} X_{1,t}\right] - \mathbb{P}(I_T \neq 1)\mathbb{E}\left[\max_{t\leq T} X_{1,t}\right] .
\end{aligned}$$

The fourth equality holds because the fact that arm k is chosen by the algorithm after the exploration phase is independent of the rewards that are available for arm k in the exploitation phase. We also used that as the distributions are supported on $\mathbb{R}$ the expectation of their maximum is positive for T large enough. This concludes the proof.

□

C PROOFS OF SECTION 4 (SDA)

C.1 Proof of Proposition 2

Proposition 2 (Regret decomposition with a low probability event)**.** *Define the event*

$$\xi_T := \{N_1(T) \leq T - KM_T\} \ ,$$

where $(M_T)_{T\in\mathbb{N}}$ *is a fixed sequence. Then, for* $T \geq KM_T$*, for any constant* $x_T \in \mathbb{R}$*, it holds that,*

$$\mathcal{R}_T^\pi \leq \underbrace{\mathbb{E}\left[\max_{t\leq T} X_{1,t}\right] - \mathbb{E}\left[\max_{t\leq T-KM_T} X_{1,t}\right]}_{\textit{Exploration cost}} + \underbrace{x_T\mathbb{P}(\xi_T) + \mathbb{E}\left[\max_{t\leq T} X_{1,t}\mathbb{1}\left(\max_{t\leq T} X_{1,t} \geq x_T\right)\right]}_{\textit{Cost incurred by } \xi_T} \ .$$

Proof. We recall that $N_k(t)$ denotes the number of pulls of arm k at time t.

$$\begin{aligned}
\mathbb{E}\left[\max_{t\leq T} X_{I_t,t}\right] &\geq \mathbb{E}\left[\max_{n\leq N_1(T)} X_{1,n}\right] \quad \text{(keeping observations from a single arm)} \\
&\geq \mathbb{E}\left[\max_{t\leq N_1(T)} X_{1,t}\mathbb{1}(\xi_T^c)\right] \\
&\geq \mathbb{E}\left[\max_{t\leq T-KM_T} X_{1,t}\mathbb{1}(\xi_T^c)\right] \\
&= \mathbb{E}\left[\max_{t\leq T-KM_T} X_{1,t}\right] - \mathbb{E}\left[\max_{t\leq T-KM_T} X_{1,t}\mathbb{1}(\xi_T)\right] \\
&\geq \mathbb{E}\left[\max_{t\leq T-KM_T} X_{1,t}\right] - \mathbb{E}\left[\max_{t\leq T} X_{1,t}\mathbb{1}(\xi_T)\right] \ .
\end{aligned}$$

At this step the decomposition is very similar to the one of the ETC proof. However, this time the event ξ_T is not independent on the maximum on the available rewards so we need to control the expectation more precisely. We use the notation $X_T^+ = \max_{t\leq T} X_{1,t}$ for simplicity, and then consider a constant $x_T \in \mathbb{R}$ and write

$$\begin{aligned}
\mathbb{E}\left[\max_{t\leq T} X_{1,t}\mathbb{1}(\xi_T)\right] &= \mathbb{E}[X_T^+\mathbb{1}(\xi_T)] \leq \mathbb{E}[X_T^+\mathbb{1}(\xi_T)\mathbb{1}(X_T^+ \leq x_T)] + \mathbb{E}[X_T^+\mathbb{1}(\xi_T)\mathbb{1}(X_T^+ \geq x_T)] \\
&\leq x_T\mathbb{P}(\xi_T) + \mathbb{E}[X_T^+\mathbb{1}(X_T^+ \geq x_T)] \ .
\end{aligned}$$

This concludes the proof. □

Before going further with this result, we can make a few remarks.

Remark 3 (Comparison with the regret bound for ETC strategies)**.** *The expression we obtain can be compared with the result for the ETC strategies. The first part (exploration cost) is similar, with* M_T *as the total number of samples collected during the exploration phase. The second term is more complicated as we simply had* $\mathbb{P}(\xi_T)\mathbb{E}[\max_{t\leq T} X_{1,t}]$ *for the ETC strategy. We now require this decomposition because the event* ξ_T *is correlated with all rewards from arm* 1*. However, the upper bound* $\mathbb{P}(\xi_T)\mathbb{E}[\max_{t\leq T} X_{1,t}]$ *should hold because intuitively* ξ_T *and the maximum should be negatively correlated, as* ξ_T *corresponds to arm* 1 *under-performing. This seems however very intricate to prove.*

C.2 Proof of Lemma 3

Lemma 3 (Upper bound on $\mathbb{P}(\xi_T)$)**.** *For any $q \in (0,1)$, any M_T and any $\gamma > 0$, under QoMax-SDA with parameters $B(n) = n^\gamma$ and $f(r) = (\log r)^{\frac{1}{\gamma}}$,*

$$\mathbb{P}(\xi_T) = \mathcal{O}\left((\log T)^{\frac{1}{\gamma}} M_T^{-\frac{1}{1+\gamma}}\right) .$$

Moreover, for all $k \neq 1$, $\mathbb{E}[n_k(T)] = \mathcal{O}((\log T)^{1/\gamma})$.

Proof. We recall $\xi_T := \{N_1(T) \leq T - KM_T\}$. First, using $\sum_{k=1}^K N_k(T) = T$, we remark that

$$\mathbb{P}(N_1(T) \leq T - KM_T) \leq \mathbb{P}(\exists k \geq 2, N_k(T) \geq M_T) \leq \sum_{k=2}^K \mathbb{P}(N_k(T) \geq M_T) ,$$

We denote by r_T the index of the round for which the number of observations equals or exceeds T. As at least one observation is collected at the end of the round it holds that $r_T \leq T$. Hence, we can obtain

$$\mathbb{P}(N_1(T) \leq T - KM_T) \leq \sum_{k=2}^K \mathbb{P}(n_k(r_T) b_k(r_T) \geq M_T) \leq \sum_{k=2}^K \mathbb{P}(n_k(T) b_k(T) \geq M_T) .$$

Using $b_k(T) = n_k(T)^\gamma$ and Markov inequality gives

$$\mathbb{P}(N_1(T) \leq T - KM_T) \leq \sum_{k=2}^K \mathbb{P}(n_k(T)^{1+\gamma} \geq M_T) \leq \sum_{k=2}^K \mathbb{P}(n_k(T) \geq M_T^{\frac{1}{1+\gamma}}) \leq \sum_{k=2}^K \frac{\mathbb{E}[n_k(T)]}{M_T^{\frac{1}{1+\gamma}}} .$$

For all $k \geq 2$, Lemma 5 (proved in Section C.3) shows that $\mathbb{E}[n_k(T)] = \mathcal{O}\left((\log T)^{\frac{1}{\gamma}}\right)$, with the tuning we choose for the algorithm. This concludes the proof. □

C.3 Proof of Lemma 5

The remaining part consists in upper bounding the expectation of the number of queries of each suboptimal arm for T rounds of QoMax-SDA, for which we will apply techniques similar to the proof of LB-SDA (see Baudry et al. (2021)).

Lemma 5. *Under QoMax-SDA with parameters $B(n) = n^\gamma$ and $f(r) = (\log r)^{\frac{1}{\gamma}}$, for all $k \geq 2$, there exists a constant C_k such that the number of pulls of arm k at time T satisfies*

$$\mathbb{E}[n_k(T)] \leq C_k \left(\log(T)\right)^{\frac{1}{\gamma}} + \mathcal{O}(1) .$$

Proof. In the proof, we denote the q-QoMAx from arm k using the samples between the sample n_1 and the sample n_2, $\bar{X}^q_{k,n_1:n_2}$ and the q-QoMax from arm k using the first n samples by $\bar{X}^q_{k,n}$. Note that we omit the dependency in the batch size b because this one is implicit through $B(n) = n^\gamma$.

(1) A first decomposition. We start with a decomposition similar to the one proposed for LB-SDA, which is that for any function $n_0(T)$ we have

$$\begin{aligned}
\mathbb{E}[n_k(T)] &= \mathbb{E}\left[\sum_{r=0}^{T-1} \mathbb{1}(k \in \mathcal{A}_{r+1})\right] = \mathbb{E}\left[\sum_{r=0}^{T-1} \mathbb{1}(k \in \mathcal{A}_{r+1}, \ell(r) = 1)\right] + \mathbb{E}\left[\sum_{r=1}^{T-1} \mathbb{1}(k \in \mathcal{A}_{r+1}, \ell(r) \neq 1)\right] \\
&\leq \mathbb{E}\left[\sum_{r=0}^{T-1} \mathbb{1}(k \in \mathcal{A}_{r+1}, \ell(r) = 1, n_k(r) \leq n_0(T))\right] + \mathbb{E}\left[\sum_{r=0}^{T-1} \mathbb{1}(k \in \mathcal{A}_{r+1}, \ell(r) = 1, n_k(r) \geq n_0(T))\right] \\
&\quad + \quad \mathbb{E}\left[\sum_{r=1}^{T-1} \mathbb{1}(\ell(r) \neq 1)\right] \\
&\leq n_0(T) + \mathbb{E}\left[\underbrace{\sum_{r=0}^{T-1} \mathbb{1}(k \in \mathcal{A}_{r+1}, \ell(r) = 1, n_k(r) \geq n_0(T)}_{A}\right] + \mathbb{E}\left[\sum_{r=1}^{T-1} \mathbb{1}(\ell(r) \neq 1)\right] ,
\end{aligned}$$

where we used that

$$\begin{aligned}
\sum_{r=0}^{T-1} \mathbb{1}(k \in \mathcal{A}_{r+1}, n_k(r) \leq n_0(T)) &\leq \sum_{r=0}^{T-1} \mathbb{1}(k \in \mathcal{A}_{r+1}, n_k(r) \leq n_0(T)) \\
&\leq \sum_{r=0}^{T-1} \sum_{n=1}^{n_0(T)} \mathbb{1}(k \in \mathcal{A}_{r+1}, \ell(r) = 1, n_k(r) = n) \\
&\leq \sum_{n=1}^{n_0(T)} \left(\sum_{r=0}^{T-1} \mathbb{1}(k \in \mathcal{A}_{r+1}, \ell(r) = 1, n_k(r) = n)\right) \\
&\leq \sum_{n=1}^{n_0(T)} 1 = n_0(T) ,
\end{aligned}$$

as the event $\{k \in \mathcal{A}_{r+1}, n_k(r) = n\}$ can only happen at one round.

(2) Upper bound for A. Now, we can upper bound the counterpart with $n_k(r) \geq n_0(T)$, using the concentration from Theorem 1.

$$\begin{aligned}
A &\coloneqq \mathbb{E}\left[\sum_{r=0}^{T-1} \mathbb{1}(k \in \mathcal{A}_{r+1}, \ell(r) = 1, n_k(r) \geq n_0(T))\right] \\
&\leq \mathbb{E}\left[\sum_{r=0}^{T-1} \mathbb{1}(k \in \mathcal{A}_{r+1}, \bar{X}^q_{k,n_k(r)} \geq \bar{X}^q_{1,n_1(r)-n_k(r)+1:n_1(r)}, \ell(r) = 1, n_k(r) \geq n_0(T))\right] \\
&\leq \mathbb{E}\left[\sum_{r=0}^{T-1} \mathbb{1}\left(\bar{X}^q_{k,n_k(r)} \geq x_{n_k(r)}, n_k(r) \geq n_0(T), k \in \mathcal{A}_{r+1}\right)\right] \\
&\quad + \quad \mathbb{E}\left[\sum_{r=0}^{T-1} \mathbb{1}\left(\bar{X}^q_{1,n_1(r)-n_k(r)+1:n_1(r)} \leq x_{n_k(r)}, \ell(r) = 1, n_k(r) \geq n_0(T), k \in \mathcal{A}_{r+1}\right)\right] ,
\end{aligned}$$

where we used that if the QoMax of k exceeds the QoMax of 1, then it is either larger than x_n or the QoMax of 1 is smaller than x_n for any arbitrary choice of x_n. In our case, we will choose a convenient value of x_n to use Theorem 1. Using union bounds on the number of queries it then holds that

$$\begin{aligned}
A &\leq \mathbb{E}\left[\sum_{r=0}^{T-1} \sum_{n_k=n_0(T)}^{T-1} \mathbb{1}(\bar{X}^q_{k,n_k} \geq x_{n_k}, k \in \mathcal{A}_{r+1}, n_k(r) = n_k)\right] \\
&\quad + \quad \mathbb{E}\left[\sum_{r=0}^{T-1} \sum_{n_k=n_0(T)}^{T-1} \sum_{n=r/K}^{T-1} \mathbb{1}(\bar{X}^q_{1,n-n_k+1:n} \leq x_{n_k}, \ell(r) = 1, n_k(r) = n_k, k \in \mathcal{A}_{r+1}, n_1(r) = n)\right] .
\end{aligned}$$

We now use the same trick as before to reduce the double sum on r and n_k to only one sum, and write that

$$\begin{aligned}
A &\leq \mathbb{E}\left[\sum_{n_k=n_0(T)}^{T-1} \mathbb{1}(\bar{X}^q_{k,n_k} \geq x_{n_k}) \sum_{r=0}^{T-1} \mathbb{1}(k \in \mathcal{A}_{r+1}, n_k(r) = n_k)\right] + \mathbb{E}\left[\sum_{n_k=n_0(T)}^{T-1} \sum_{n=r/K}^{T-1} \mathbb{1}(\bar{X}^q_{1,n-n_k+1:n} \leq x_{n_k})\right] \\
&\leq \sum_{n_k=n_0(T)}^{T-1} \mathbb{P}(\bar{X}^q_{k,n_k} \geq x_{n_k}) + \sum_{n_k=n_0(T)}^{T-1} \sum_{n=r/K}^{T-1} \mathbb{P}(\bar{X}^q_{1,n-n_k+1:n} \leq x_{n_k}) \\
&\leq \sum_{n_k=n_0(T)}^{T-1} \mathbb{P}(\bar{X}^q_{k,n_k} \geq x_{n_k}) + T \sum_{n_k=n_0(T)}^{T-1} \mathbb{P}(\bar{X}^q_{1,n_k} \leq x_{n_k}) \, .
\end{aligned}$$

Plugging the concentration result from Theorem 1, one has

$$A \leq \sum_{n_k=n_0(T)}^{T-1} e^{-c_k b(n_k)} + T \sum_{n=n_0(T)}^{T-1} e^{-c_1 b(n)} \leq T e^{-c_k b(n_0(T))} + T^2 \exp(-c_1 b(n_0(T))) \, .$$

Let n_0 the integer for which Theorem 1 can be applied between the arm 1 and any arm k for $k \geq 2$. Now we choose, $n_0(T) = \max\left(b^{-1}\left(\frac{2\log T}{c_1}\right), b^{-1}\left(\frac{\log T}{c_k}\right), n_0\right)$. With this choice, we get

$$A = \mathbb{E}\left[\sum_{r=0}^{T-1} \mathbb{1}(k \in \mathcal{A}_{r+1}, \ell(r) = 1, n_k(r) \geq n_0(T))\right] = \mathcal{O}(1) \, . \tag{6}$$

If we define $b(n) = n^\gamma$, using Equation (6) and the decomposition for $\mathbb{E}[n_k(T)]$, it holds that for some constant C

$$\mathbb{E}[n_k(T)] \leq C(\log(T))^{\frac{1}{\gamma}} + \mathbb{E}\left[\sum_{r=0}^{T-1} \mathbb{1}(\ell(r) \neq 1)\right] + \mathcal{O}(1) \, . \tag{7}$$

The next step of the proof is to have a deeper look at $\mathbb{E}\left[\sum_{r=0}^{T-1} \mathbb{1}(\ell(r) \neq 1)\right]$.

(3) Upper bound for $\mathbb{E}\left[\sum_{r=0}^{T-1} \mathbb{1}(\ell(r) \neq 1)\right]$. We provide a similar decomposition as in Baudry et al. (2021), considering the case where arm 1 has already been leader and the alternative. Before that we recall the following property obtained by the definition of the leader

$$\ell(r) = k \Rightarrow n_k(r) \geq \left\lceil \frac{r}{K} \right\rceil \, .$$

We then define $a_r = \left\lceil \frac{r}{4} \right\rceil$, and write that

$$\mathbb{P}\left(\ell(r) \neq 1\right) = \mathbb{P}\left(\{\ell(r) \neq 1\} \cap \mathcal{D}^r\right) + \mathbb{P}\left(\{\ell(r) \neq 1\} \cap \bar{\mathcal{D}}^r\right) \, , \tag{8}$$

where we define $\mathcal{D}^r$ the event under which the asymptotically dominating arm has been leader at least once in $[a_r, r]$.

$$\mathcal{D}^r = \{\exists u \in [a_r, r] \text{ such that } \ell(u) = 1\}.$$

We now explain how to upper bound the term in the left hand side of Equation (8). We look at the rounds larger than some round r_0 that will be specified later in the proof.

We introduce a new event

$$\mathcal{B}^u = \{\ell(u) = 1, k \in \mathcal{A}_{u+1}, n_k(u) = n_1(u) \text{ for some arm } k\} \, .$$

Under the event $\mathcal{D}^r$, $\{\ell(r) \neq 1\}$ can only be true if the leadership has been taken over by a suboptimal arm at some round between a_r and r, that is

$$\{\ell(r) \neq 1\} \cap \mathcal{D}^r \subset \cup_{u=a_r}^{r-1} \{\ell(u) = 1, \ell(u+1) \neq 1\} \subset \cup_{u=a_r}^{r} \mathcal{B}^u \ . \tag{9}$$

This is because a leadership takeover can only happen after a challenger has defeated the leader while having the same number of observations. Moreover, each leadership takeover has been caused by either (1) a QoMax of a challenger is over-performing, or (2) a QoMax of the leader is under-performing, with a sample size in each case larger than $s_r = \lceil a_r/K \rceil$. In addition, each of these QoMax can **only cost one takeover** (thanks to the subsampling scheme), hence we can simply use an union bound on these events. In summary, after defining some $r_0 > 8$ we have that

$$\begin{aligned}
\mathbb{E}\left[\sum_{r=0}^{T-1} \mathbb{1}(\ell(r) \neq 1, \mathcal{D}_r)\right] &\leq r_0 + \mathbb{E}\left[\sum_{r=r_0}^{T-1}\left(\sum_{u=a_r}^{r} \mathbb{1}(\mathcal{B}^u)\right)\right] \\
&\leq r_0 + \mathbb{E}\left[\sum_{r=r_0}^{T-1}\left(\sum_{n=s_r}^{r}\left(\mathbb{1}(\bar{X}^q_{1,n} \leq x_n) + \sum_{k=2}^{K} \mathbb{1}(\bar{X}^q_{k,n} \geq x_n)\right)\right)\right] \\
&\leq r_0 + \sum_{r=r_0}^{T-1}\sum_{n=s_r}^{r}\left(\mathbb{P}(\bar{X}^q_{1,n} \leq x_n) + \sum_{k=2}^{K} \mathbb{P}(\bar{X}^q_{k,n} \geq x_n)\right) \\
&\leq r_0 + \sum_{k=1}^{K}\sum_{r=r_0}^{T-1}\sum_{n=s_r}^{r} \exp(-c_k b(n)) \\
&\leq r_0 + \sum_{k=1}^{K}\sum_{r=r_0}^{T-1} r\exp(-c_k b(s_r)) \\
&= \mathcal{O}(1) \ ,
\end{aligned}$$

since $b(s_r) = s_r^\gamma = \Omega(r^\gamma)$. This is true if $s_{r_0} \geq n_0$, which is the condition that allows the use of the concentration inequality from Theorem 1. We consider r_0 large enough to satisfy this condition.

We now handle the case when the asymptotically dominant arm has never been leader between a_r and r, which implies that it has lost a lot of duels against the respective leaders of many rounds. We introduce

$$\mathcal{L}^r = \sum_{u=a_r}^{r} \mathbb{1}(\mathcal{C}^u) \ ,$$

with $\mathcal{C}^u = \{\exists k \neq 1, \ell(u) = k, 1 \notin \mathcal{A}_{u+1}\}$. It is proved in Chan (2020) that

$$\mathbb{P}(\ell(r) \neq 1 \cap \bar{\mathcal{D}}^r) \leq \mathbb{P}(\mathcal{L}^r \geq r/4) \ . \tag{10}$$

and the author uses the Markov inequality to provide the upper bound

$$\mathbb{P}(\mathcal{L}^r \geq r/4) \leq \frac{\mathbb{E}(\mathcal{L}^r)}{r/4} = \frac{4}{r}\sum_{u=a_r}^{r} \mathbb{P}(\mathcal{C}^u) \ . \tag{11}$$

From this step we can refactor $\sum_{r=r_0}^{r} \mathbb{P}(\mathcal{L}^r \geq r/4)$ using the following trick from Baudry et al. (2020),

$$\begin{aligned}
\sum_{r=r_0}^{T-1} \frac{4}{r}\mathbb{1}(u \in [a_r, r]) &= \sum_{r=r_0}^{T-1} 4\frac{\mathbb{1}(u \leq r)}{r}\mathbb{1}(a_r \leq u) \leq \frac{4}{u}\sum_{r=r_0}^{T-1} \mathbb{1}(a_r \leq u) \\
&\leq \frac{4}{u}\sum_{r=r_0}^{T-1} \mathbb{1}(\lceil r/4 \rceil \leq u) \leq \frac{4}{u}\sum_{r=r_0}^{T-1} \mathbb{1}(r/4 \leq u+1) \\
&\leq \frac{4}{u} \times 4(u+1) \leq 32 \ .
\end{aligned}$$

With this result we obtain that

$$\sum_{r=r_0}^{T-1} \mathbb{P}(\ell(r) \neq 1 \cap \bar{\mathcal{D}}^r) \leq \sum_{r=r_0}^{T-1} \mathbb{P}(\mathcal{L}^r \geq r/4) \leq 32 \sum_{r=a_{r_0}}^{T-1} \mathbb{P}(\mathcal{C}^r) \, .$$

Now we can have a more precise look at $\mathbb{P}(\mathcal{C}^r) = \mathbb{P}(\exists k \neq 1, \ell(r) = k, 1 \notin \mathcal{A}_{r+1})$. We recall that we defined in the algorithm a forced exploration $f(r)$, ensuring that $n_k(r) \geq f(r)$ for any arm k and any round r.

$$\begin{aligned}
\sum_{r=a_{r_0}}^{T-1} \mathbb{P}(\mathcal{C}^r) &\leq \sum_{r=a_{r_0}}^{T-1} \mathbb{P}\left(\{\bar{X}^q_{1,n_1(r)} \leq x_{n_1(r)}\} \cup_{k=2}^K \left\{\bar{X}^q_{k,n_k(r)-n_1(r)+1:n_1(r)} \geq x_{n_1(r)}, \ell(r) = k\right\}\right) \\
&\leq \sum_{r=a_{r_0}}^{T-1} \sum_{n=f(r)}^{r/2} \mathbb{P}(\bar{X}^q_{1,n} \leq x_n) + \sum_{k=2}^K \sum_{r=a_{r_0}}^{T-1} \sum_{n=f(r)}^{r/2} \sum_{n_k=\lceil r/K \rceil}^{r} \mathbb{P}(\bar{X}^q_{k,n_k-n+1:n_k} \geq x_n) \\
&\leq \sum_{r=a_{r_0}}^{T-1} \sum_{n=f(r)}^{r/2} \exp(-c_1 b(n)) + \sum_{k=2}^K \sum_{r=a_{r_0}}^{T-1} \sum_{n=f(r)}^{r/2} r \exp(-c_k b(n)) \\
&\leq \sum_{r=a_{r_0}}^{T-1} \frac{r}{2} \exp(-c_1 b(f(r))) + \sum_{k=2}^K \sum_{r=a_{r_0}}^{T-1} \frac{r^2}{2} \exp(-c_k b(f(r))) \, ,
\end{aligned}$$

where the use of the concentration from Theorem 1 is permitted only if $f(a_{r_0}) \geq n_0$. Now, this result provides a sound theoretical tuning for the forced exploration parameter as a function of b, as choosing $f(r) \geq \max_k \left(\frac{4 \log r}{c_k}\right)^{\frac{1}{\gamma}}$ ensures

$$\sum_{r=a_{r_0}}^{T-1} \mathbb{P}(\mathcal{C}^r) = \mathcal{O}(1) \, .$$

Hence, we obtain the final result that for some constant C_k it holds that

$$\mathbb{E}[n_k(T)] \leq C_k (\log(T))^{\frac{1}{\gamma}} + \mathcal{O}(1) \, ,$$

under the assumptions that Theorem 1 can be applied and that the forced exploration is of the same scaling as the regret, namely $f(r) = \Omega((\log r)^{\frac{1}{\gamma}})$. □

C.4 Proof of Theorem 3

Theorem 3 (Upper bound on the regret of QoMax-SDA)**.** *For any quantile q, any $\gamma > 0$, defining the parameters of QoMax-SDA as $B(n) = n^\gamma$ and $f(r) = (\log r)^{\frac{1}{\gamma}}$. The regret of QoMax-SDA is **(1)** vanishing in the strong sense for exponential tails **(2)** vanishing in the weak sense for polynomial tails.*

Proof. We instantiate the decomposition of Proposition 2 using the value of $\mathbb{P}(\xi_T)$ obtained in Lemma 3. Plugging all of these values and using similar tricks as those already used in Appendix A.4 to establish Assumption 1 for semi-parametric tails, we write for π being any instance of QoMax-SDA with parameter γ,

$$\mathcal{R}_T^\pi \leq \underbrace{\mathbb{E}\left[\max_{t\leq T} X_{1,t}\right] - \mathbb{E}\left[\max_{t\leq T-KM_T} X_{1,t}\right]}_{\text{Exploration cost}} + \underbrace{x_T \mathbb{P}(\xi_T) + \mathbb{E}\left[\max_{t\leq T} X_{1,t} \mathbb{1}\left(\max_{t\leq T} X_{1,t} \geq x_T\right)\right]}_{\text{Cost incurred by } \xi_T}$$
$$\leq \underbrace{KM_T\left[\frac{B_T}{T} + \int_{B_T}^{+\infty} G_1(x)dx\right]}_{(A_1)} + \underbrace{x_T \frac{C(\log T)^{\frac{1}{\gamma}}}{M_T^{\frac{1}{1+\gamma}}}}_{(A_2)} + \underbrace{T\int_{x_T}^{+\infty} G_1(x)dx}_{(A_3)} ,$$

for any values of x_T, B_T, M_T, that we now specify for each of the two families considered.

Exponential tails We recall that if $G_1(x) = \mathcal{O}(\exp(-\lambda x))$, then for any $y \in \mathbb{R}$ we have

$$\int_y^{+\infty} G_1(x)dx = \mathcal{O}\left(\exp(-\lambda y)\right) .$$

First, if we choose $B_T = \frac{1}{\lambda}\log(T)$ then (A_1) vanishes for any choice of $M_T = T^\alpha$ with $0 < \alpha < 1$. Similarly, choosing $x_T = \frac{2}{\lambda}\log T$ ensures that $(A_3) = \mathcal{O}(1/T)$. Then, (A_2) is in $\mathcal{O}\left(\frac{(\log T)^{1+\frac{1}{\gamma}}}{M_T^{\frac{1}{1+\gamma}}}\right)$, which is vanishing for any choice of $M_T = T^\alpha$, $\alpha \in (0,1)$. We conclude that for exponential tails, $\lim_{T\to\infty} \mathcal{R}_T^\pi = 0$.

Polynomial tails Consider again $M_T = T^\alpha$, for some $\alpha \in (0,1)$. This time,

$$\int_y^{+\infty} G_1(x)dx = \mathcal{O}\left(\frac{1}{y^{\lambda-1}}\right) .$$

Plugging into (A_3), we get a term of order $\mathcal{O}(T \times x_T^{1-\lambda})$. Let's take $x_T = T^\beta$ for some $\beta \in (0,1)$, we then have

$$(A_3) = \mathcal{O}(T^{1+\beta(1-\lambda)}) .$$

Now consider (A_2), omitting the polylog terms we obtain

$$(A_2) = \mathcal{O}(T^{\beta - \frac{\alpha}{1+\gamma}}) .$$

Consider finally (A_1). Choosing $B_T = T^{\frac{1}{\lambda}}$ (as in Appendix A.4) we obtain the tightest upper bound on the exploration cost:

$$(A_1) = \mathcal{O}\left(\frac{M_T}{T^{1-\frac{1}{\lambda}}}\right) = \mathcal{O}(T^{\alpha - 1 + \frac{1}{\lambda}}) .$$

To get the smallest order with this proof technique we want to equalize all these three exponents, which gives

$$\alpha - 1 + \frac{1}{\lambda} = \beta - \frac{\alpha}{1+\gamma} = 1 + \beta(1-\lambda) .$$

For simplicity we write $\beta = \frac{1}{\lambda} + \eta$ and try to find η instead. Re-writing the the three equalities yields

$$\alpha + \frac{1}{\lambda} - 1 = \frac{1}{\lambda} + \eta - \frac{\alpha}{1+\gamma} = \frac{1}{\lambda} - (\lambda - 1)\eta .$$

This can be further simplified in

$$\alpha - 1 = \eta - \frac{\alpha}{1+\gamma} = -(\lambda - 1)\eta .$$

This gives in particular a system of two equations with two unknowns η and α. By substituing α we get

$$\begin{aligned} \eta - \frac{1-(\lambda-1)\eta}{1+\gamma} &= -(\lambda-1)\eta \\ \Leftrightarrow \eta\left[1+\gamma+\lambda-1+(\lambda-1)(1+\gamma)\right] &= 1\,, \end{aligned}$$

which gives $\eta = \frac{1}{\lambda(2+\gamma)-1}$ and $\alpha = \frac{\lambda(1+\gamma)}{\lambda(2+\gamma)-1}$.

Plugging in these values, we obtain that (A_1), (A_2) and (A_3) are all in $\mathcal{O}\left(T^{\frac{1}{\lambda}-\frac{\lambda-1}{\lambda(2+\gamma)-1}}\right) = o(T^{1/\lambda})$. Recalling the rate of growth of the maximum for polynomial tails given in Proposition 3 we get that for polynomial tails

$$\mathcal{R}_T^{\pi} = \underset{T\to\infty}{o}\left(\mathbb{E}\left[\max_{t\leq T} X_{1,t}\right]\right).$$

□

C.5 Possible Mistake in the Analysis of Max-Median

In the proof of Theorem 4.1 of Bhatt et al. (2021) the authors upper bound

$$P\left(m(n) \geq v_n, W_i(n) \geq (1+\delta)\lambda_i^{-1}\log(m(n))\right)$$

where $v_n = \frac{1}{a}\sum_{d=1}^{n}\varepsilon_d$, $m(n) = \min_k N_k(n)$ and $W_i(n)$ is the index used by Max-Median for arm i, which is the order statics of order $\lfloor N_i(n)/m(n)\rfloor$. To do so, they use union bounds and concentration of a binomial random variable, which can be rewritten as follows:

$$\begin{aligned} P\left(m(n) \geq v_n, W_i(n) \geq (1+\delta)\lambda_i^{-1}\log(m(n))\right) &\leq \sum_{m\geq v_n}\sum_{k\geq m}\mathbb{P}\left(\mathcal{O}_{i,k}\left(\left\lfloor\frac{k}{m}\right\rfloor\right) \geq (1+\delta)\lambda_i^{-1}\log(m)\right) \\ &\leq \sum_{m\geq v_n}\sum_{k\geq m}\mathbb{P}\left(S_k \geq \frac{k}{m}\right)\,, \end{aligned}$$

where S_k counts the number of observations among the k first observations from arm i that are exceeding $(1+\delta)\lambda_i^{-1}\log(m)$. From the tail assumption, S_k is a binomial distribution with parameter k and $p = a_i/m^{1+\delta}$.

To upper bound this last probability, the authors use an exponential Markov inequality with a particular value of θ. Using instead Chernoff inequality, which consists in optimizing over θ to get the smallest possible upper bound, one obtains

$$\mathbb{P}\left(S_k \geq \frac{k}{m}\right) = \mathbb{P}\left(\frac{S_k}{k} \geq \frac{1}{m}\right) \leq \exp\left(-k\mathrm{kl}\left(\frac{1}{m}, \frac{a_i}{m^{1+\delta}}\right)\right),$$

provided that k/m exceeds the mean $a_i/m^{1+\delta}$, where $\mathrm{kl}(x,y) = x\log(x/y) + (1-x)\log((1-x)/(1-y))$ is the binary relative entropy. Hence, for n large enough,

$$P\left(m(n) \geq v_n, W_i(n) \geq (1+\delta)\lambda_i^{-1}\log(m(n))\right) \leq \sum_{m\geq v_n}\sum_{k\geq m}\exp\left(-k\mathrm{kl}\left(\frac{1}{m}, \frac{a_i}{m^{1+\delta}}\right)\right).$$

In the proof of Theorem 4.1, Bhatt et al. (2021) end up summing a quantity that does not depend on m, $\exp(-k^{\delta}/2)$, but it seems to be obtained by mistaking k/m by m in the tail probability of the binomial distribution. Without this mistake and with the tightest possible bound on the tail of a binomial distribution, the upper bound we obtain does depend on m. More precisely as

$$\mathrm{kl}\left(\frac{1}{m}, \frac{a_i}{m^{1+\delta}}\right) \sim \frac{\delta\log(m)}{m}\,,$$

when m tends to infinity, one obtains an upper bound of order

$$B_n = \sum_{m\geq v_n}\sum_{k\geq m}\exp\left(-k\frac{\delta\log(m)}{m}\right) = \sum_{m\geq v_n}\frac{\exp\left(-m\frac{\delta\log(m)}{m}\right)}{1-\exp\left(-\frac{\delta\log(m)}{m}\right)}\,.$$

Given that

$$\frac{\exp\left(-m\frac{\delta\log(m)}{m}\right)}{1-\exp\left(-\frac{\delta\log(m)}{m}\right)} \sim \frac{m^{1-\delta}}{\delta\log(m)} \ ,$$

when m tends to infinity, we don't see how we can get $\sum_n B_n < \infty$ (for any small enough δ) which is needed in the rest of the proof of Theorem 4.1 in order to be able to apply Borel Cantelli's lemma.

D COMPLEMENTS OF SECTION 5: PRACTICAL PERFORMANCE OF QOMAX ALGORITHMS

D.1 Implementation Tricks for QoMax-SDA

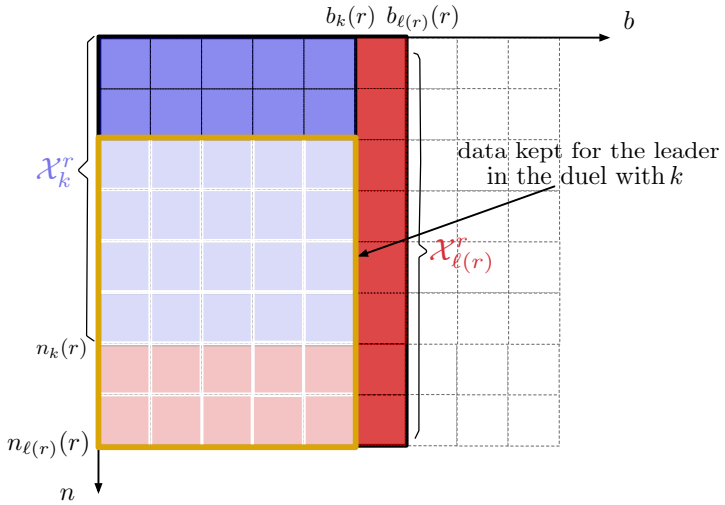

Figure 3: Illustration of the subsampling mechanism used by QoMax-SDA between the leader and a challenger k at round r.

In this section we detail the CollectData procedure used for QoMax-SDA (see Algorithm 1) and briefly introduced in Section 4. In particular, in Algorithm 5 we describe the second implementation trick that allows to reduce significantly the memory complexity of QoMax-SDA. The principle is actually quite simple: the "last-block" subsampling that considers a subsample of the leader's history of the same size as the challenger's to compute their QoMax will take the *last queries* for the leader as illustrated on Figure 3. Moreover, as we look for the maximum of this subsample on each batch, it is clear that we can remove a lot of information. For instance, imagine that the last data pulled for a batch is its global maximum. Then all previously stored data in this batch is useless when looking at the last block and can be deleted. If we apply this principle recursively, we have the following: (1) the newly pulled observation is necessarily kept (in case we consider a last block of size 1), (2) we can remove *all observations smaller than this last data*: again, looking at these samples in the past will not change anything for the value of the maximum. To implement this trick we store a list of data and a list of their indexes (i.e to which query of the arm they correspond) in order to know where to look at when performing the subsampling step for the leader. More than a storage trick, this is also time efficient: the global maximum of the batch is simply the first element of the list, and for a subsample of batches between queries $n_\ell - n_k$ and n_ℓ the subsample maximum is simply the observation corresponding to the first index in $\mathcal{I}_\ell$ (defined in Algorithm 6) larger than $n_\ell - n_k$.

Details of the procedure CollectData. Considering Algorithm 5 for the addition of new data, we can summarize the procedure in very few steps: (1) For each arm k that is queried, **add one observation to each of its existing batches**. (2) For each queried arm k that is *not the leader*, **collect as many batches as necessary** to match the number of batches $B(n_k)$. (3) Collect as many batches as necessary for the leader to match the number of batches of the arm with the second largest number of queries.

Algorithm 5 Efficient Update of a list of maxima for QoMax-SDA

Input: List of indices $\mathcal{I} = \{i_1, \dots, i_L\}$, sorted list $\mathcal{X} = \{X_1, \dots, X_L\}$, $X_1 > X_2 > \cdots > X_L$, new index i, new data X
Search step: Find the largest j satisfying $X_j > X$ (Binary Search)
Update step:
Set $\mathcal{X} \leftarrow \{X_1, X_2, \dots, X_j, X\}$ // Remove $X_{j+1}, \dots, X_L$ and add X
Set $\mathcal{I} \leftarrow \{i_1, \dots, i_j, i\}$ // Remove $i_{j+1}, \dots, i_L$ and add i
Return: List of indices $\mathcal{I}$, list of data $\mathcal{X}$.

Algorithm 6 Collect Data at the end of a round for QoMax-SDA

Input: K arms with data $\mathcal{X}_k$ stored as $(\mathcal{I}_k^{(j)}, \mathcal{X}_k^{(j)})_{j\in\{1,\dots,b_k\}}$, number of queries n_k for each arm, distributions $(\nu_k)_{k\in K}$, $\mathcal{A}$: set of arms chosen by QoMax-SDA, ℓ current leader, B function controlling the batch size

for $k \in \{1,\dots,K,\ell\}$ **do**
  if $b_k > 0$, $k \in \mathcal{A}$ **then**
    $n_k \leftarrow n_k + 1$ // Update the number of queries of arm k
    for $j \in \{1,\dots,b_k\}$ **do**
      Collect $X \sim \nu_k$ (Update batch j) // Add one observation in each existing batch j of $\mathcal{X}_k$
      $(\mathcal{I}_k^{(j)}, \mathcal{X}_k^{(j)}) \leftarrow$ EfficientUpdate$(\mathcal{I}_k^{(j)}, \mathcal{X}_k^{(j)}, n_k, X)$ (Alg. 5)
  if $k \neq \ell$ **then**
    $B_{\text{new}} = B(n_k)$ // New batch size computed with B if k is a challenger.
  else
    $B_{\text{new}} = \max_{k\neq\ell} b_k$ // If k is leader, align its batch size to the second most pulled challenger.
  while $b_k \leq B_{new}$ **do**
    $\mathcal{I}_k^{(b_k+1)}, \mathcal{X}_k^{(b_k+1)} \leftarrow \{\}, \{\}$
    for $i \in \{1,\dots,n_k\}$ **do**
      Collect $X \sim \nu_k$ // Collect n_k data to form a new batch
      $(\mathcal{I}_k^{(b_k+1)}, \mathcal{X}_k^{(b_k+1)}) \leftarrow$ EfficientUpdate$(\mathcal{I}_k^{(b_k+1)}, \mathcal{X}_k^{(b_k+1)}, i, X)$ (Alg. 5)
    Add $(\mathcal{I}_k^{(b_k+1)}, \mathcal{X}_k^{(b_k+1)})$ in $\mathcal{X}_k$
    $b_k \leftarrow b_k + 1$

Return: $\mathcal{X}_1, \dots, \mathcal{X}_K$

Empirical evidences of the efficiency of the storage trick (Algorithm 5). We propose simulations to verify that the solution we propose to store the data used by QoMAx-SDA is indeed efficient. We performed 1000 simulations for each sample size $N \in [10^2, 5\times10^2, 10^3, 2\times10^3, 5\times10^3, 10^4, 2\times10^4, 3\times10^4, 5\times10^4])$, and for 4 distributions: (1) a Pareto distribution with tail parameter 1.1, (2) a Pareto distribution with tail parameter 3, (3) an exponential distribution with parameter 1, (4) a standard normal distribution. We report in Figure 4 the average number of data stored by the algorithm for each sample sizes, along with the empirical 5% and 95% quantiles on the 1000 simulations and the curve $y = \log(N)$ for comparison. We observe that: (1) The results do not depend on the distribution. (2) All 4 curves are very close to exactly $\log(N)$, which is as small as ≈ 10 for a sample size of 5×10^4. (3) 90% of the simulations have no more than 17 data stored, and the maximum we observe on all 4 experiments is actually 23 which is very small compared to $N = 5\times10^4$.

Therefore, we conclude that the trick we introduced is indeed efficient and our experiments corroborate the intuition that it allows to store $\mathcal{O}(\log N)$ data out of N on average. We now prove it formally in Lemma 6

Lemma 6 (Expected memory with the efficient storing of maxima)**.** *Denote by C_N the random variable denoting the memory usage of a random i.i.d sample of size N drawn from any distribution with the implementation trick from Alg. 5. For any ν, it holds that*

$$\mathbb{E}[C_N] = \sum_{n=1}^{N} \frac{1}{n} \sim \log(N) \,.$$

Proof. Denote the sorted random samples by $X_1 > \cdots > X_N$. As the observations are i.i.d, all of them are equally likely to be in the last position. We consider I_N the random variable denoting the index of the last observation, it holds that

$$\mathbb{E}[C_N] = \frac{1}{N}\sum_{j=1}^{N} \mathbb{E}[C_N | I_N = j] \,.$$

Then, we remark that if $I_N = j$, all observations of higher order $X_{j+1}, \dots, X_N$ are removed from the history. Hence, it only remains to count the average amount of data considering only $X_1, \dots, X_{j-1}$, which is equal to $\mathbb{E}[C_{j-1}]$ and add 1 for the last observation. Using that $\mathbb{E}[C_1] = 1$,

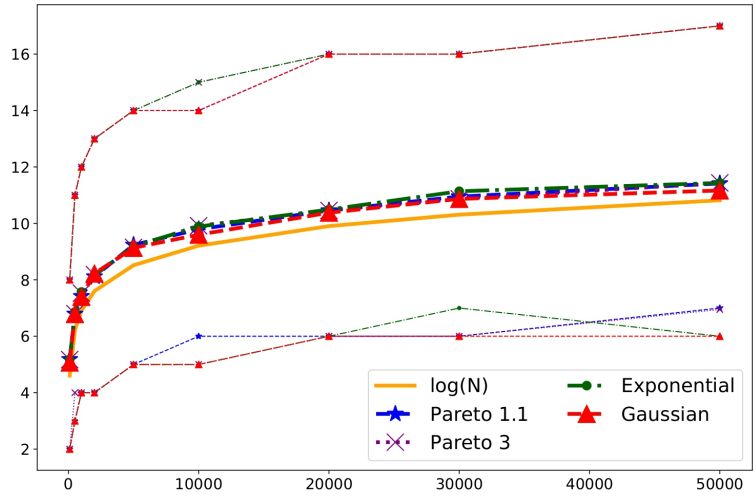

Figure 4: Average number of data kept in memory with the efficient storage of maxima, for 1000 simulations with sample size $N \in [10^2, 5\times10^2, 10^3, 2\times10^3, 5\times10^3, 10^4, 2\times10^4, 3\times10^4, 5\times10^4]$ and the empirical 5% and 95% quantiles.

$$\mathbb{E}[C_N] = \frac{1}{N}\sum_{j=1}^{N}\mathbb{E}[C_N|I_N=j] = \frac{1}{N}\sum_{j=1}^{N}(1+\mathbb{E}[C_{j-1}])$$
$$\Rightarrow (N+1)\mathbb{E}[C_{N+1}] - N\mathbb{E}[C_N] = \sum_{j=1}^{N+1}(1+\mathbb{E}[C_j]) - \sum_{j=1}^{N}(1+\mathbb{E}[C_j]) = 1+\mathbb{E}[C_N]$$
$$\Rightarrow (N+1)(\mathbb{E}[C_{N+1}] - \mathbb{E}[C_N]) = 1$$
$$\Rightarrow \mathbb{E}[C_{N+1}] = \mathbb{E}[C_N] + \frac{1}{N+1}$$
$$\Rightarrow \mathbb{E}[C_N] = \sum_{n=1}^{N}\frac{1}{n}\,.$$

□

D.2 More on the Storage/Computation Time

In this section we detail the computation of the storage constraints and time complexities reported in Table 1. We restate them in Table 2 and express them as a function of their parameters.

ThresholdAscent (Streeter and Smith, 2006b). After simplifying the statement of the algorithm presented in their paper (beginning of Section 3) for continuous distributions, we find out that the algorithm actually considers the *s largest observations observed so far* where s is a parameter of the algorithm. Indeed as long as the threshold is larger than s observations, the threshold is increased. The authors suggest taking $s = 100$, implying a memory complexity as small as 100 observations. After remarking this, the implementation of the algorithm is simplified largely (see our code): we can drop all data that are not in the s largest observed so far, and the index needs to be re-computed only if this list changes. Asymptotically we expect this list to change very rarely, hence the time complexity of the algorithm is dominated by the check that an observation is larger than the s-th largest reward collected so far.

ExtremeHunter/ExtremeETC (Carpentier and Valko, 2014; Achab et al., 2017). The values we provide for these algorithms in Table 2 come directly from Achab et al. (2017). We recall that in Table 1 we considered $b = 1$, but we remark that even with b very large ($b = +\infty$ corresponds to exact Pareto distributions) the memory and time complexities cannot go below $K(\log T)^2$ and $K(\log T)^4$ respectively.

Table 2: Average time and storage complexities of Extreme Bandit algorithms according to their parameters for a time horizon T.

Algorithm	Memory usage	Time complexity
ThresholdAscent	s	$\mathcal{O}(KT)$
Extreme Hunter	T	$\mathcal{O}(T^2)$
MaxMedian	T	$\mathcal{O}(KT\log T)$
QoMax-SDA	$\mathcal{O}((\log T)^2 + (K-1)\log(T)\log\log(T))$	$\mathcal{O}(KT\log T)$
Extreme ETC	$K(\log T)^{2+\frac{1}{b}}$	$\mathcal{O}\left(K(\log T)^{2\times\left(2+\frac{1}{b}\right)}\right)$
QoMax-ETC	Kb_T	$\mathcal{O}\left(\max\left\{Kb_Tn_T, Kb_T\log(b_T)\right\}\right)$

MaxMedian (Bhatt et al., 2021). In short, MaxMedian essentially tracks the quantile of order $1/m(t)$ for each arm, where $m(t)$ is the number of samples from the arm that has been pulled the least. Technically, even if it is unlikely, all observations could be used in the future (if we continuously collect data that are smaller than all values we obtained so far). For this reason, all T observations have to be stored.

QoMax-ETC (this paper). The storage required is simply Kb_T, which corresponds to storing online the maximum of each batch (b_T batches) for each arm. Collecting the b_T maxima takes a $\mathcal{O}(K\times n_T\times b_T)$ time (collecting an observation and comparing it with a current maximum costs $\mathcal{O}(1)$). At the final step of the exploration phase, computing the QoMax takes an additional $\mathcal{O}(Kb_T\log b_T)$, which is the cost of sorting K lists of size b_T. So, as a function of (n_T, b_T), the complexity of the algorithm is $\mathcal{O}\left(\max\left\{b_T\times n_T, b_T\log b_T\right\}\right)$.

QoMax-SDA (this paper). We recall that QoMax-SDA uses a batch size n^γ for an arm that has been queried n times, a forced exploration $(\log r)^{1/\gamma}$, and that the total number of queries of every sub-optimal arm is provably $\mathcal{O}((\log T)^{1/\gamma})$ (see Theorem 2). We start with the computation of the memory capacity and detail how the two implementation tricks presented in D.1 work: (1) indexing the number of batches of the leader to the second most pulled arm allows to reduce the number of batches of the leader from T^γ to $\mathcal{O}\left(\left((\log T)^{1/\gamma}\right)^\gamma\right) = \mathcal{O}(\log T)$, for any γ. Then, we look at how many data are stored in each batch, and (2) according to Lemma 6 the efficient storage of maxima allows to store only $\mathcal{O}(\log N)$ observations out of N on average. This gives $\mathcal{O}(\log T)$ for the leader, and $\mathcal{O}(\log\log T)$ for the challengers. This explains why the dependency of K in the memory becomes a second order term for T large enough. Then, we consider the computational time, which can be divided into two steps that are executed at each round: (a) updating the lists of values (each batch of the K arms), and (b) computing the $K-1$ QoMax for the challengers and the $K-1$ QoMax for the leader. Operation (a) requires to find the index from which previous data can be erased. As the list is sorted (by construction), this can take up to $(\log N)$ with N the sample size of a batch using a binary search. Hence, this gives $\mathcal{O}(\log\log T)$ for each batch of the leader and $\mathcal{O}(\log\log\log T)$ for each batch of the challengers. On the other hand, for step (b) the efficient storage ensures that we have access to the maximum of each batch at constant cost (first observation of the list), and we only need to find the quantiles over the different batches, giving $\mathcal{O}(2(K-1)\log T)$. Hence, we can report an overall $\mathcal{O}(KT\log T)$ time complexity, or a $\mathcal{O}(T\log T\log\log T)$ when T is very large. We report the first, because if $K=5$ then $\log\log T > K$ only for $T > 10^{65}$, which is unreasonably large.

D.3 Supplementary for Section 5 : Additional Experiments

In this section we provide the complete results for all the experiments we performed and that were advertised in Section 5. We first reproduce the experiments from previous papers, and then consider a few new settings. Before that, we detail the parameters used for each algorithm.

The code to reproduce the experiments is available on Github.

Parameters for All Experiments. We recall the parameters we used for the different experiments. For each experiment, we run $N = 10^4$ independent trajectories for 10 time horizons $T \in [1000, 2500, 5000, 7500, 9000, 10000, 15000, 20000, 30000, 50000]$. This methodology is computationally expensive

but allows for a fair comparison between ETC and more adaptive strategies. Furthermore, it is also a way to stabilize the results because if the same trajectories were used to plot the results for different time horizons then a few extreme trajectories for some algorithms would have too much influence on our conclusions. This is not a problem as all runs for $T \leq 20000$ are actually quite fast with parallel computing, and the total computation time of our experiments is largely dominated by the experiment with $T = 50000$.

The parameters we used are the following:

- ThresholdAscent: $s = 100, \delta = 0.1$, as suggested in Streeter and Smith (2006b).
- ExtremeETC/ExtremeHunter: $b = 1$, as in Carpentier and Valko (2014). As the authors, we use $\delta = 0.1$ for the experiments instead of the theoretical value that is too large for the time horizons considered, and $D = E = 10^{-3}$ for the UCB. Other theoretically-motivated parameters are $r = T^{-1/(2b+1)}$ (fraction of samples used for the tail estimation), $N = (\log T)^{\frac{2b+1}{b}}$ (length of the initial exploration phase). $\delta = \exp(-\log^2(T))/(2TK)$ in the paper but set to 0.1 here.
- MaxMedian: The exploration probability is set to $\varepsilon_t = 1/(1+t)$ as suggested in Bhatt et al. (2021).
- QoMax-ETC: We test $q = 1/2$ and $q = 0.9$, $b_T = (\log T)^2$ and $n_T = \log T$ to match both the theoretical requirements of Section 3 and the length of the exploration phase of ExtremeETC for a fair comparison.
- QoMax-SDA: $f(r) = (\log r)^{\frac{1}{\gamma}}$ and $B(n) = n^\gamma$ for $\gamma = 2/3$, which works well across all the experiments we performed. The quantile is either equal to $q = 1/2$ or $q = 0.9$.

D.3.1 Experiments 1-6

We describe the setting of each experiment, that we will then refer by their number (e.g exp.1).

1. (exp.1 in Bhatt et al. (2021)): $K = 5$ Pareto distributions with tail parameters $\lambda_k \in [2.1, 2.3, 1.3, 1.1, 1.9]$.
2. (exp.2 in Bhatt et al. (2021)) $K = 7$ Pareto distributions with $\lambda_k \in [2.5, 2.8, 4, 3, 1.4, 1.4, 1.9]$. All arms have a scaling $C = 1$ except arm 5 with $C_5 = 1.1$. Hence ν_5 is the dominating arm from a slight margin.
3. (exp.3 in Bhatt et al. (2021)) $K = 10$ Exponential arms with a survival function $G_k(x) = e^{-\lambda_k x}$ with parameters $\lambda_k = [2.1, 2.4, 1.9, 1.3, 1.1, 2.9, 1.5, 2.2, 2.6, 1.4]$.
4. (exp.4 in Bhatt et al. (2021)) $K = 20$ Gaussian arms, with same mean $\mu_k = 1, \forall k$, and different variances $\sigma_k = [1.64, 2.29, 1.79, 2.67, 1.70, 1.36, 1.90, 2.19, 0.80, 0.12, 1.65, 1.19, 1.88, 0.89, 3.35, 1.5, 2.22, 3.03, 1.08, 0.48]$. The dominant arm has a standard deviation 3.35.
5. (exp.1 in Carpentier and Valko (2014)) $K = 3$ Pareto distributions with $\lambda \in [5, 1.1, 2]$.
6. (exp.2 in Carpentier and Valko (2014)) $K = 3$ arms, including 2 Pareto distributions with $\lambda_k \in [1.5, 3]$, and arm 3 is a mixture Dirac/Pareto: pull 0 with 80% probability, reward from a Pareto distribution with $\lambda = 1.1$ with 20% probability. Hence, the last arm dominates asymptotically.

Objective of each experiment. Before reporting the results, we explain why each experiment is interesting in our opinion for the empirical evaluation of Extreme Bandits algorithms. Experiment 1 is quite difficult because the tail gap between arm 3 and arm 4 is relatively small.Otherwise, all algorithms are supposed to have guarantees in this setting so their comparison is fair. Experiment 2 allows to consider a semi-parametric setting with a tail gap $\delta_{5-6} = 0$, hence it only holds that $\nu_5 \succ \nu_6$: we check whether the algorithms are able to (1) pull 5 and 6 most often, and (2) arbitrate in favor of arm 5. Then, experiments 3 and 4 allow to test the different algorithms respectively with exponential and gaussian tails (with different variances), showing the performance of the algorithms when the tails are not polynomial. Moreover, a larger number of arms is considered in these experiments. Finally, experiment 5 is relatively easy and more of a sanity check for the performance of each algorithm (it was exp.1 in Carpentier and Valko (2014)). Experiment 6 will be interesting for discussing the limits of parameter-free approaches, as the dominant tail provides low rewards with relatively high probability.

Results For each experiment, we report the results according to the criteria **(I)-(IV)** that are introduced in Section 5. The criteria **(I)-(II)** are reported side by side for each experiment in Figures 5-10. Tables 3-13 associated with **(III)** report the result for the statistics on the number of pulls of the best arm on all trajectories at $T = 5 \times 10^4$. Finally, Tables 4-14 related to **(IV)** report the results for the statistics on the empirical distribution of the maxima on all trajectories at $T = 5 \times 10^4$.

We summarize our key observations on the results with the following points:

- **On the non-robustness of reporting the average maximum collected.** Several examples can serve to illustrate this point. For experiment 1 (Table 4) if we look at the average maximum only, we would conclude that QoMax-SDA with $q = 1/2$ is by far the best algorithm with an average of 1.8×10^5 (1.1×10^5 for the second). However, we see that the quantiles of the maxima distributions are almost identical to those of other QoMax algorithms. Hence, even if 99% of their distribution matches, QoMax-SDA with $q = 1/2$ has a nearly 70% better average caused by less than 1% of the trajectories. The same thing seems to happen on different problems: the 10^4 and 8.5×10^3 of 1/2-QoMax-ETC and ExtremeETC are clearly over-estimated means in experiment 2 considering that they both have the same quantiles as 1/2-QoMax-SDA (even a bit worse), which has an average of 7.5×10^3, and MaxMedian with 7.9×10^3. This variability is even more striking in Experiment 5 (see Table 12) where ExtremeETC has three times the average maximum of ExtremeHunter. Without surprise, this phenomenon is more present when the tails are heavier. Hence looking at the average maxima is meaningful with the statistics from Experiments 3 and 4 with lighter tails.

- **Quantiles.** We recall that for metric **(I)** we use a quantile to estimate the expectation of the maximum, $\tilde{q} = \mathbb{P}(X_T^+ \leq \mathbb{E}[X_T^+]) \approx \exp(-TG(\mathbb{E}[X_T^+]))$. In the experiments we plug the equivalents of $\mathbb{E}[X_T^+]$ in each setting: for Pareto distribution we obtain $\tilde{q} = \exp\left(-\frac{1}{\Gamma(1-1/\lambda)^\lambda}\right)$, for exponential we obtain $\tilde{q} = e^{-1}$, and for Gaussian distributions we compute the value numerically (c.f notebook provided with the code).

- **QoMax Performance.** QoMax algorithms clearly outperform their competitors in Experiments **1, 3, 4** and **5** according to all criteria. As those experiments include polynomial, exponential and gaussian tails with different number of arms, this shows the generality and efficiency of the QoMax approach. QoMax-SDA seems to work better than QoMax-ETC, in particular it is competitive even for small time horizons ($T < 5 \times 10^3$) in most experiments. However, we see that QoMax-ETC almost matches the performance of QoMax-SDA for $T = 5 \times 10^4$. For a practitioner who would be interested in larger time horizons QoMax-ETC seems to be a perfectly suitable choice.

- On the contrary, **ExtremeHunter** performs significantly better than **ExtremeETC** for larger horizons: the probability of mistake of the latter is still quite large, and the ability of ExtremeHunter to recover from a mistake is valuable, but we recall that the time complexity of ExtremeHunter is detrimental for the practitioner. Results from Experiments **3** and **4** show that the two algorithms are not able to handle exponential and gaussian tails.

- **ThresholdAscent** is never the best algorithm but has the advantage of being consistently better than the uniform strategy (according to **(II)**), as it always pulls the best arm at a frequency larger that $1/K$. It is the most stable baseline in terms of **(III)** (it always has the narrowest range for the statistics we consider), but this is detrimental to its capacity to collect large values.

- We tested **MaxMedian** on larger time horizons than in the original paper, which explains the difference in some results. Indeed, we observe that in Experiments **1, 3, 5**, MaxMedian is quite competitive for shorter time horizons ($T \leq 10^4$), but almost stops improving at this step. This suggests that the algorithm does not explore enough, which is confirmed by a closer look at **(III)**: the number of pulls of the best arm are either very close to 0% or to 100% in most of the cases, which is a behavior specific to this algorithm and that we would like to avoid in practice. This behavior also has an impact on the statistics on the maxima distributions **(IV)**. The exploration function may be partly responsible for this : in Experiment **4** MaxMedian fails with the Gaussians[4], and actually commits to the *worst* arm at an early stage. Indeed, with 20 arms and $\varepsilon_t = 1/(t+1)$ we are very likely to have at least one arm that is never sampled twice, while this is the case the order statistics used is always the *minimum*, favoring the arm with the *lowest* variance instead of the largest. We think that a deterministic forced exploration could at least partially solve this.

[4] which is not what Bhatt et al. (2021) obtained, but we were not able to find why we could not reproduce their results.

- In **Experiment 2** MaxMedian performs very well and commits very early to the best arm for most of the trajectories. QoMax algorithms are clearly slower, but still pull the best arm more than 50% of the time. Moreover, when we add the number of pulls of the second best arm we get around 90% for all QoMax algorithms, which is clearly competitive. Indeed, the empirical regret of both QoMax-SDA (Figure 6 (left)) is close to the one of MaxMedian, and we see in Table 6 that their quantiles of the maxima distributions are also very close to the one of MaxMedian. Hence, we think that QoMax-SDA may have chosen more often the second best arm when it provided very large rewards, which is not a problem according to the initial objective of the algorithms.
- **Experiment 6** shows that in some examples parametric algorithms can perform much better than non-parametric approaches. Indeed, the distribution of arm 3 enters in the second-order Pareto family, and the parameter $b = 1$ makes ExtremeHunter calibrate its parameters with the $\approx 5\%$ best samples of each arm. This is enough for the algorithm to "detect" the Pareto tail of the mixture and sample it most often. Most of the other algorithms fail, including QoMax, to the exception of ThresholdAscent which still pulls the best arm 40% of the time at $T = 5 \times 10^4$. However, this experiment also illustrates two important remarks on QoMax: the 0.9-QoMax-SDA performs much better than the others, showing that when the tails are harder to detect choosing a larger quantile can be valuable. Furthermore, we tested another experiment imposing at least 100 samples in each batch. This time, 0.9-QoMax-SDA was able to pull the best arm 60% of the time. Hence, this gives the practitioner the ability to increase the exploration and the quantile q if very difficult tails are expected, which depends on the characteristics of the real problem at hand.

Considering all these points, we think that QoMax-ETC and QoMax-SDA are very practical solutions in addition to their strong theoretical guarantees. They work well on most examples **with the same parameters** (avoiding painful tuning), including settings with different kind of tails (polynomial, exponential, gaussian) with different number of arms, and both easy and hard instances. We saw however with experiment 6 the limits of a distribution-free approach if we consider a hard problem. It also showed that in this case augmenting the quantile q (and/or the forced exploration function f for QoMax-SDA) used in QoMax algorithms can be beneficial. Furthermore, we can recommend to use QoMax-ETC when the time horizon will be very large (larger than 5×10^4 for instance) and QoMax-SDA for smaller time horizons, as it seems to learn faster on all examples but is more computationally demanding.

D.3.2 Experiments with Log-Normal and Generalized Gaussian Distributions

In this section we add two new experiments, considering two new families of distributions: (1) the log-normal distribution (2 parameters (μ, σ), if X follows a log-normal distribution with these parameters then $\log(X) \sim \mathcal{N}(\mu, \sigma)$), and (2) the generalized normal distribution (a parameter β and a density $\sim \exp\left(-|x|^\beta\right)$).

- **Experiment 7:** We consider $K = 5$ log-normal arms with parameters $\mu_k \in [1, 1.5, 2, 3, 3.5]$ and $\sigma_k \in [4, 3, 2, 1, 0.5]$. When T is large enough the parameter σ determines which arm dominates (arm 1 in our case).
- **Experiment 8:** We consider $K = 8$ generalized gaussian arms with parameters $\beta_k \in (0.2 \times i)_{i\in\{1,\dots,8\}}$. Hence, the heavier tail is arm 1.

We run the same algorithms as for Experiments 1-6, with the exact same parameters for all of them. This time we cannot report **(I)** because we cannot compute the proxy empirical regret. Hence, we report in Figure 11 and Figure 12 the number of pulls of the dominant arm for the two experiments, along with the statistics corresponding to evaluation criteria **(III)-(IV)** for these experiments. These two additional experiments further highlight the generality and performance of QoMax algorithms compared to the other Extreme Bandits baselines.

Experiment 1

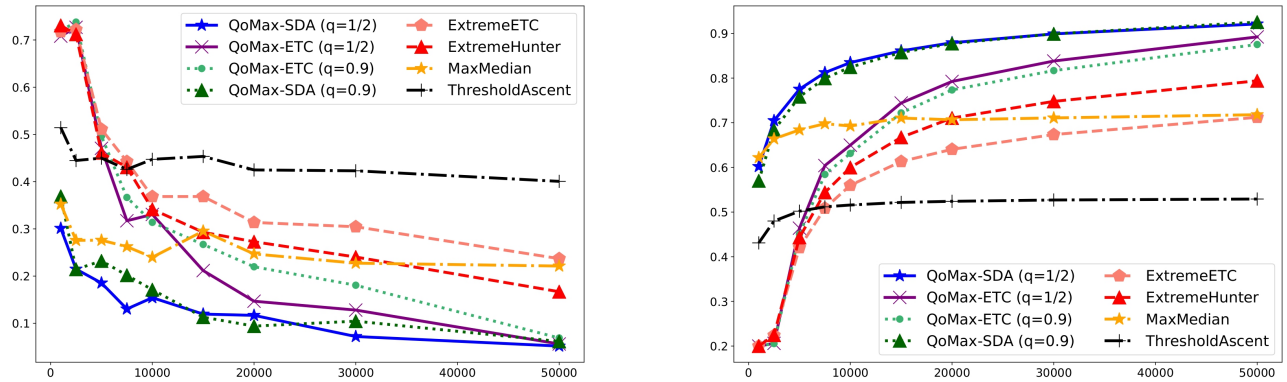

Figure 5: Experiment 1: Proxy Empirical Regret (left) and Number of pulls of the dominant arm (right), averaged over 10^4 independent trajectories for $T \in \{10^3, 2.5 \times 10^3, 5 \times 10^3, 7.5 \times 10^3, 9 \times 10, 10^4, 1.5 \times 10^4, 2 \times 10^4, 3 \times 10^4, 5 \times 10^4\}$.

Table 3: Statistics on the number of pulls of the best arm at $T = 5 \times 10^4$, Experiment 1.

Algorithm	Average (%)	1%	10%	25%	50%	75%	90%	99%
QoMax-SDA ($q = 1/2$)	92	42	90	93	94	95	95	95
QoMax-SDA ($q = 0.9$)	93	14	87	93	96	97	98	98
QoMax-ETC ($q = 1/2$)	89	90	90	90	90	90	90	90
QoMax-ETC ($q = 0.9$)	88	3	90	90	90	90	90	90
ExtremeETC	71	3	3	90	90	90	90	90
ExtremeHunter	79	3	5	89	90	90	90	90
MaxMedian	72	0	0	0	100	100	100	100
ThresholdAscent	53	46	50	52	53	55	56	57

Table 4: Statistics on the distributions of maxima at $T = 5 \times 10^4$, Experiment 1. Results divided by 100 to improve readability.

Algorithm	Average	1%	10%	25%	50%	75%	90%	99%
QoMax-SDA ($q = 1/2$)	1852	41	81	130	245	547	1350	11371
QoMax-SDA ($q = 0.9$)	1042	39	78	128	239	529	1363	12539
QoMax-ETC ($q = 1/2$)	1058	40	79	126	232	530	1324	11054
QoMax-ETC ($q = 0.9$)	919	34	75	122	230	511	1301	10080
ExtremeETC	882	16	44	86	183	426	1089	9515
ExtremeHunter	1092	21	61	104	208	477	1226	9799
MaxMedian	785	3	37	83	180	436	1126	9240
ThresholdAscent	748	27	51	82	156	351	853	7771

Experiment 2

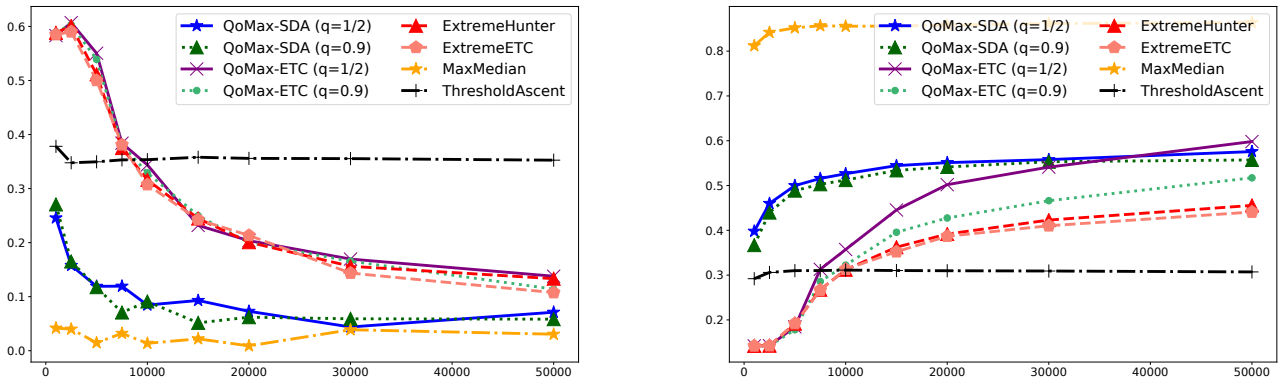

Figure 6: Experiment 2: Proxy Empirical Regret (left) and Number of pulls of the dominant arm (right), averaged over 10^4 independent trajectories for $T \in \{10^3, 2.5 \times 10^3, 5 \times 10^3, 7.5 \times 10^3, 9 \times 10, 10^4, 1.5 \times 10^4, 2 \times 10^4, 3 \times 10^4, 5 \times 10^4\}$.

Table 5: Statistics on the number of pulls of the best arm at $T = 5 \times 10^4$, Experiment 2.

Algorithm	Average (%)	1%	10%	25%	50%	75%	90%	99%
QoMax-SDA ($q = 1/2$)	58	2	6	23	72	88	91	92
QoMax-SDA ($q = 0.9$)	56	1	6	21	66	88	94	97
QoMax-ETC ($q = 1/2$)	60	3	3	3	84	84	84	84
QoMax-ETC ($q = 0.9$)	52	3	3	3	84	84	84	84
ExtremeETC	44	3	3	3	85	85	85	85
ExtremeHunter	46	3	3	3	71	85	85	85
MaxMedian	86	0	0	100	100	100	100	100
ThresholdAscent	31	20	25	28	31	34	36	40

Table 6: Statistics on the distributions of maxima at $T = 5 \times 10^4$, Experiment 2. Results divided by 100 to improve readability.

Algorithm	Average	1%	10%	25%	50%	75%	90%	99%
QoMax-SDA ($q = 1/2$)	75	8	13	18	30	56	112	657
QoMax-SDA ($q = 0.9$)	69	8	13	18	30	56	113	614
QoMax-ETC ($q = 1/2$)	98	7	12	17	28	51	105	616
QoMax-ETC ($q = 0.9$)	65	7	12	17	28	52	107	564
ExtremeETC	85	5	12	17	28	53	108	638
ExtremeHunter	61	7	12	17	28	52	100	522
MaxMedian	79	6	13	19	31	57	116	664
ThresholdAscent	46	5	9	13	21	38	77	418

Experiment 3

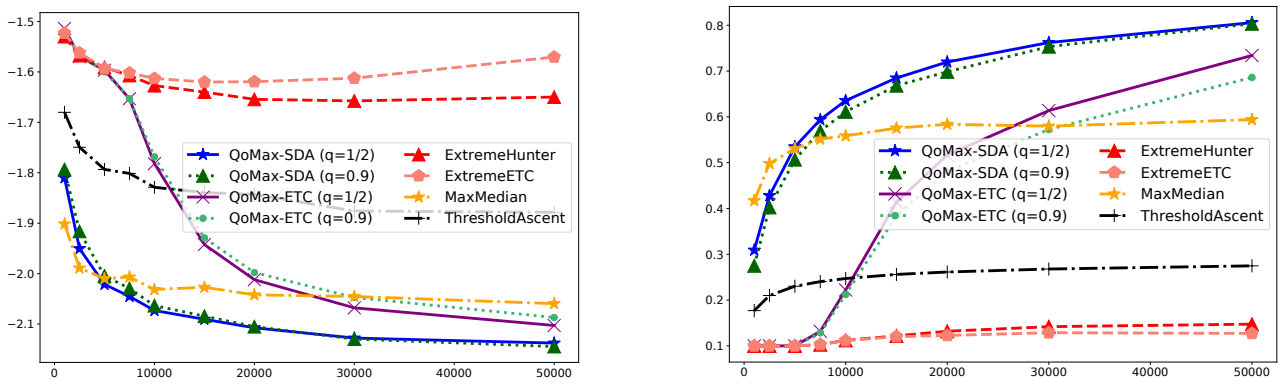

Figure 7: Experiment 3: Proxy Empirical Regret (left) and Number of pulls of the dominant arm (right), averaged over 10^4 independent trajectories for $T \in \{10^3, 2.5 \times 10^3, 5 \times 10^3, 7.5 \times 10^3, 9 \times 10, 10^4, 1.5 \times 10^4, 2 \times 10^4, 3 \times 10^4, 5 \times 10^4\}$.

Table 7: Statistics on the number of pulls of the best arm at $T = 5 \times 10^4$, Experiment 3.

Algorithm	Average (%)	1%	10%	25%	50%	75%	90%	99%
QoMax-SDA ($q = 1/2$)	81	2	72	82	86	88	88	89
QoMax-SDA ($q = 0.9$)	80	2	59	80	87	91	93	95
QoMax-ETC ($q = 1/2$)	73	3	77	77	77	77	77	77
QoMax-ETC ($q = 0.9$)	69	3	3	77	77	77	77	77
ExtremeETC	13	3	3	3	3	3	77	77
ExtremeHunter	15	3	3	3	3	7	67	77
MaxMedian	59	0	0	0	98	100	100	100
ThresholdAscent	27	21	24	26	28	29	31	33

Table 8: Statistics on the distributions of maxima at $T = 5 \times 10^4$, Experiment 3.

Algorithm	Average	1%	10%	25%	50%	75%	90%	99%
QoMax-SDA ($q = 1/2$)	32	26	28	29	31	34	37	43
QoMax-SDA ($q = 0.9$)	32	25	28	29	31	34	37	44
QoMax-ETC ($q = 1/2$)	32	25	28	29	31	34	36	43
QoMax-ETC ($q = 0.9$)	31	24	27	29	31	33	36	43
ExtremeETC	26	18	21	23	25	29	32	39
ExtremeHunter	27	19	22	23	26	29	32	39
MaxMedian	31	21	25	28	31	33	36	43
ThresholdAscent	29	23	25	27	29	31	34	41

Experiment 4

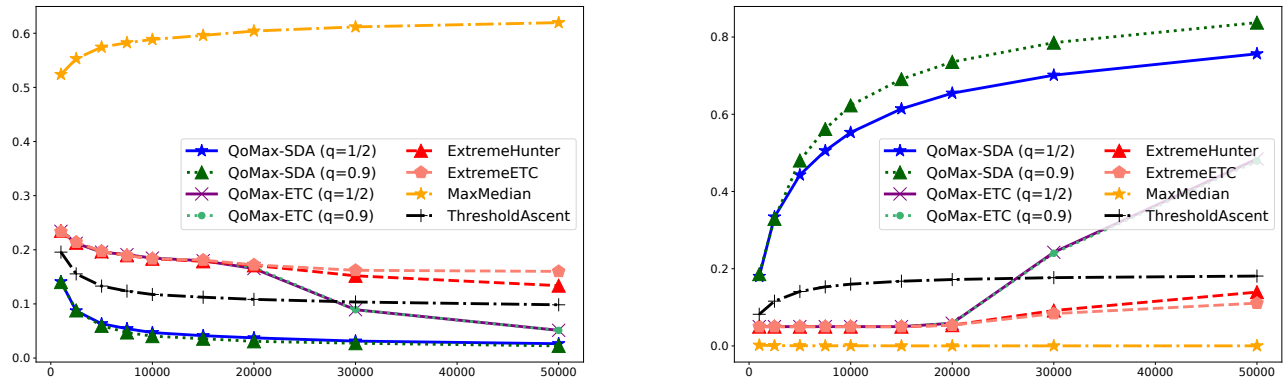

Figure 8: Experiment 4: Proxy Empirical Regret (left) and Number of pulls of the dominant arm (right), averaged over 10^4 independent trajectories for $T \in \{10^3, 2.5 \times 10^3, 5 \times 10^3, 7.5 \times 10^3, 9 \times 10, 10^4, 1.5 \times 10^4, 2 \times 10^4, 3 \times 10^4, 5 \times 10^4\}$.

Table 9: Statistics on the number of pulls of the best arm at $T = 5 \times 10^4$, Experiment 4.

Algorithm	Average (%)	1%	10%	25%	50%	75%	90%	99%
QoMax-SDA ($q = 1/2$)	76	4	74	77	78	79	80	80
QoMax-SDA ($q = 0.9$)	84	3	75	85	89	90	91	91
QoMax-ETC ($q = 1/2$)	48	3	51	51	51	51	51	51
QoMax-ETC ($q = 0.9$)	48	3	51	51	51	51	51	51
ExtremeETC	11	3	3	3	3	3	52	52
ExtremeHunter	14	3	3	3	3	25	48	52
MaxMedian	0	0	0	0	0	0	0	0
ThresholdAscent	18	15	16	17	18	19	20	20

Table 10: Statistics on the distributions of maxima at $T = 5 \times 10^4$, Experiment 4.

Algorithm	Average	1%	10%	25%	50%	75%	90%	99%
QoMax-SDA ($q = 1/2$)	14	13	13	14	14	15	16	17
QoMax-SDA ($q = 0.9$)	15	13	13	14	14	15	16	17
QoMax-ETC ($q = 1/2$)	14	12	13	13	14	15	15	17
QoMax-ETC ($q = 0.9$)	14	12	13	13	14	15	15	17
ExtremeETC	12	10	11	11	12	13	14	16
ExtremeHunter	13	10	11	12	13	14	14	16
MaxMedian	5	3	4	4	5	6	7	10
ThresholdAscent	13	12	12	13	13	14	15	16

Experiment 5

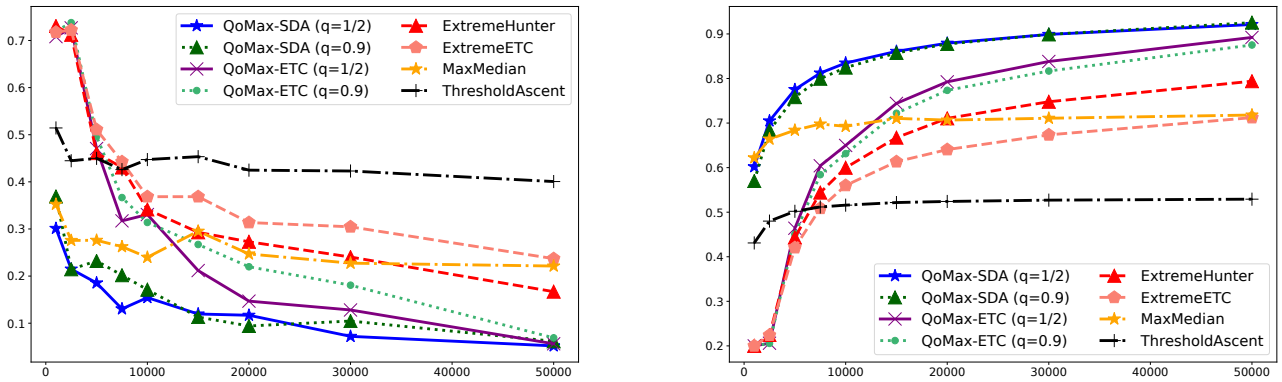

Figure 9: Experiment 5: Proxy Empirical Regret (left) and Number of pulls of the dominant arm (right), averaged over 10^4 independent trajectories for $T \in \{10^3, 2.5 \times 10^3, 5 \times 10^3, 7.5 \times 10^3, 9 \times 10, 10^4, 1.5 \times 10^4, 2 \times 10^4, 3 \times 10^4, 5 \times 10^4\}$.

Table 11: Statistics on the number of pulls of the best arm at $T = 5 \times 10^4$, Experiment 5.

Algorithm	Average (%)	1%	10%	25%	50%	75%	90%	99%
QoMax-SDA ($q = 1/2$)	97	97	97	97	97	97	97	97
QoMax-SDA ($q = 0.9$)	99	98	99	99	99	99	99	99
QoMax-ETC ($q = 1/2$)	95	95	95	95	95	95	95	95
QoMax-ETC ($q = 0.9$)	95	95	95	95	95	95	95	95
ExtremeETC	95	95	95	95	95	95	95	95
ExtremeHunter	95	95	95	95	95	95	95	95
MaxMedian	95	0	100	100	100	100	100	100
ThresholdAscent	74	73	73	74	74	74	74	74

Table 12: Statistics on the distributions of maxima at $T = 5 \times 10^4$, Experiment 5. Results divided by 100 to improve readability.

Algorithm	Average	1%	10%	25%	50%	75%	90%	99%
QoMax-SDA ($q = 1/2$)	1179	46	84	133	251	556	1405	11239
QoMax-SDA ($q = 0.9$)	1325	47	88	140	267	582	1444	12836
QoMax-ETC ($q = 1/2$)	1055	45	84	134	250	565	1347	11434
QoMax-ETC ($q = 0.9$)	944	43	82	133	247	547	1395	11038
ExtremeETC	3428	42	83	132	245	542	1362	9944
ExtremeHunter	910	44	83	132	241	553	1386	11555
MaxMedian	939	2	70	124	240	548	1378	10239
ThresholdAscent	1096	35	67	107	200	445	1151	9105

Experiment 6

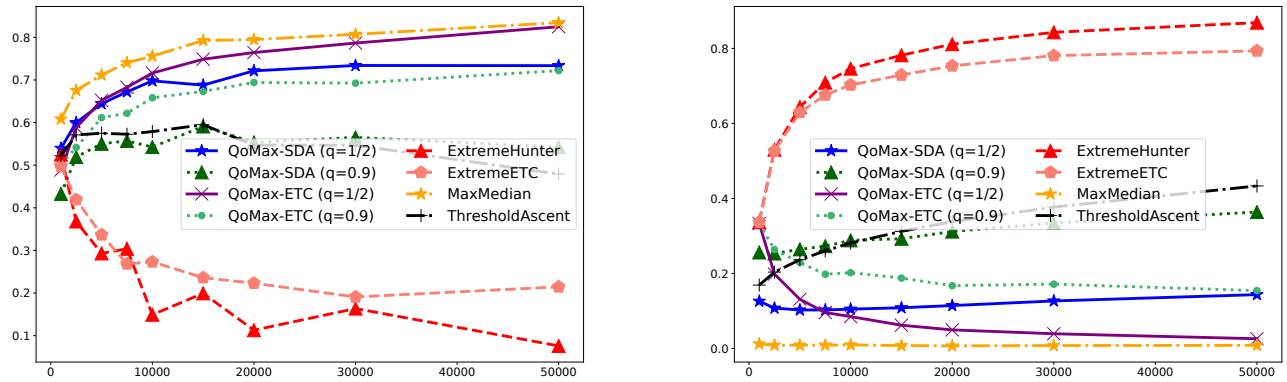

Figure 10: Experiment 6: Proxy Empirical Regret (left) and Number of pulls of the dominant arm (right), averaged over 10^4 independent trajectories for $T \in \{10^3, 2.5 \times 10^3, 5 \times 10^3, 7.5 \times 10^3, 9 \times 10, 10^4, 1.5 \times 10^4, 2 \times 10^4, 3 \times 10^4, 5 \times 10^4\}$.

Table 13: Statistics on the number of pulls of the best arm at $T = 5 \times 10^4$, Experiment 6.

Algorithm	Average (%)	1%	10%	25%	50%	75%	90%	99%
QoMax-SDA ($q = 1/2$)	14	1	1	2	4	15	45	95
QoMax-SDA ($q = 0.9$)	36	0	1	3	22	75	90	98
QoMax-ETC ($q = 1/2$)	3	3	3	3	3	3	3	3
QoMax-ETC ($q = 0.9$)	15	3	3	3	3	3	95	95
ExtremeETC	79	3	3	95	95	95	95	95
ExtremeHunter	87	3	85	95	95	95	95	95
MaxMedian	1	0	0	0	0	0	0	5
ThresholdAscent	43	27	34	38	43	49	53	60

Table 14: Statistics on the distributions of maxima at $T = 5 \times 10^4$, Experiment 6. Results divided by 100 to improve readability.

Algorithm	Average	1%	10%	25%	50%	75%	90%	99%
QoMax-SDA ($q = 1/2$)	60	5	9	12	21	41	91	635
QoMax-SDA ($q = 0.9$)	120	6	10	15	28	64	155	1144
QoMax-ETC ($q = 1/2$)	40	5	8	11	18	33	64	306
QoMax-ETC ($q = 0.9$)	59	5	8	12	20	40	93	702
ExtremeETC	267	6	14	24	47	108	266	2687
ExtremeHunter	232	8	17	28	53	116	305	2620
MaxMedian	35	0	7	10	17	30	60	306
ThresholdAscent	136	7	12	18	33	70	170	1299

Experiment 7

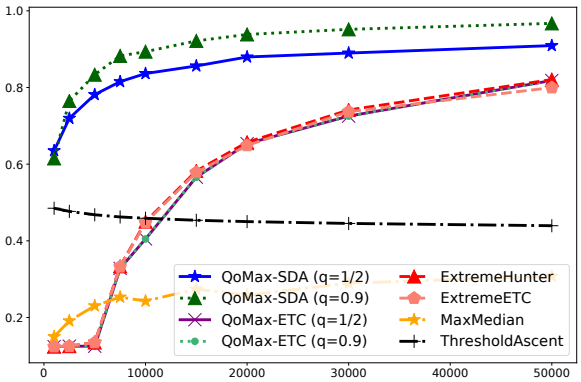

Figure 11: Experiment 7 (Log-normal arms): Number of pulls of the dominant arm, averaged over 10^4 independent trajectories for $T \in \{10^3, 2.5 \times 10^3, 5 \times 10^3, 7.5 \times 10^3, 9 \times 10, 10^4, 1.5 \times 10^4, 2 \times 10^4, 3 \times 10^4, 5 \times 10^4\}$.

Table 15: Statistics on the distributions of number of pulls of the best arm at $T = 5 \times 10^4$, Experiment 7.

Algorithm	Average	1%	10%	25%	50%	75%	90%	99%
QoMax-SDA ($q = 1/2$)	94	85	94	95	95	95	95	95
QoMax-SDA ($q = 0.9$)	97	89	96	97	98	98	98	98
QoMax-ETC ($q = 1/2$)	90	90	90	90	90	90	90	90
QoMax-ETC ($q = 0.9$)	90	90	90	90	90	90	90	90
ExtremeETC	55	3	3	3	90	90	90	90
ExtremeHunter	63	13	40	45	53	90	90	90
MaxMedian	7	0	0	0	0	0	0	100
ThresholdAscent	57	55	56	57	58	58	58	58

Table 16: Statistics on the distributions of maxima at $T = 5 \times 10^4$, Experiment 7. Results divided by 1000 to improve readability.

Algorithm	Average	1%	10%	25%	50%	75%	90%	99%
QoMax-SDA ($q = 1/2$)	1393	73	151	257	488	1090	2259	13764
QoMax-SDA ($q = 0.9$)	1401	79	163	260	524	1171	2830	13839
QoMax-ETC ($q = 1/2$)	1337	77	154	245	430	1007	2664	13651
QoMax-ETC ($q = 0.9$)	1459	84	150	251	461	987	2419	12654
ExtremeETC	957	6	12	30	214	581	1511	7422
ExtremeHunter	867	32	85	156	297	666	1569	10855
MaxMedian	76	0	0	0	0	0	15	1678
ThresholdAscent	1043	43	94	160	311	667	1648	10715

Experiment 8

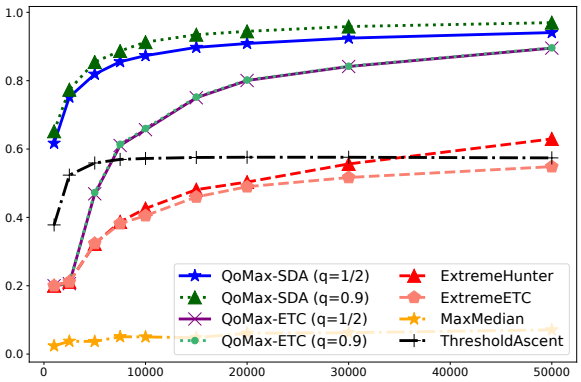

Figure 12: Experiment 8 (Generalized Gaussian arms): Number of pulls of the dominant arm, averaged over 10^4 independent trajectories for $T \in \{10^3, 2.5\times10^3, 5\times10^3, 7.5\times10^3, 9\times10, 10^4, 1.5\times10^4, 2\times10^4, 3\times10^4, 5\times10^4\}$.

Table 17: Statistics on the distributions of number of pulls of the best arm at $T = 5\times10^4$, Experiment 8.

Algorithm	Average	1%	10%	25%	50%	75%	90%	99%
QoMax-SDA ($q = 1/2$)	91	91	91	91	91	91	91	91
QoMax-SDA ($q = 0.9$)	97	97	97	97	97	97	97	97
QoMax-ETC ($q = 1/2$)	82	82	82	82	82	82	82	82
QoMax-ETC ($q = 0.9$)	82	82	82	82	82	82	82	82
ExtremeETC	80	3	82	82	82	82	82	82
ExtremeHunter	82	80	82	82	82	82	82	82
MaxMedian	31	0	0	0	0	89	100	100
ThresholdAscent	44	44	44	44	44	44	44	44

Table 18: Statistics on the distributions of maxima at $T = 5\times10^4$, Experiment 8. Results divided by 100 to improve readability.

Algorithm	Average	1%	10%	25%	50%	75%	90%	99%
QoMax-SDA ($q = 1/2$)	30	14	17	21	27	35	46	75
QoMax-SDA ($q = 0.9$)	31	14	18	21	27	34	46	94
QoMax-ETC ($q = 1/2$)	29	13	17	20	26	34	45	76
QoMax-ETC ($q = 0.9$)	29	14	17	20	26	35	45	88
ExtremeETC	28	4	17	20	25	33	43	78
ExtremeHunter	29	13	17	20	25	34	45	78
MaxMedian	11	0	0	0	0	21	33	65
ThresholdAscent	24	10	14	16	21	28	38	74