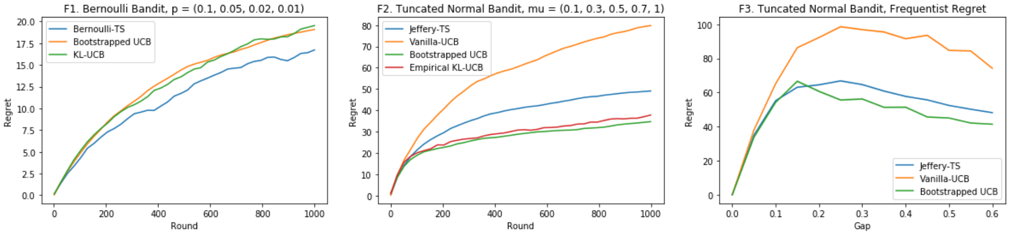
We would like to thank all reviewers for your valuable and detailed comments! We think we can fix all the issues raised by the reviewers and add more experimental results in the final version of the paper. We hope you are satisfied with our point-by-point responses and increase your scores.

Response to Reviewer 1

Experiments. Thank you for pointing out KL-UCB, a great and powerful algorithm for cases with bounded rewards. We will compare with KL-UCB in detail, and have done comprehensive experiments with KL-UCB, using Prof. Olivier Cappé's package online (due to the space limit, we can only report parts of them here). Our algorithm is comparable with KL-UCB (Bernoulli, F1) and empirical KL-UCB (truncated normal, F2) under fixed problem instances suggested in the package. Although KL-UCB is also data-dependent, our proposed method is from a very different non-parametric perspective and uses different tools by bootstrap. In practice, resampling tends to be more efficient computationally, without solving a convex optimization each round like empirical KL-UCB. Moreover, our method can work with unbounded rewards and we believe it is easier to generalize to structured bandits, e.g. linear bandit.



The UCB1 we use, defined as $\bar{y}_{n_{k,t}} + \sigma\sqrt{2\log(t)/n_{k,t}}$, exactly follows your suggestion (see E.1 in the appendix). So we will change its name to vanilla UCB. In addition, we have added the frequentist regret curve (F3). Here, the regrets of various algorithms are with respect to the instance gap (Δ) and $\mu = (\Delta, 0, \ldots, 0)$. In the linear bandit part, the dimension is specified in the title of each figure.

Complexity. Indeed, our algorithm is more complex than vanilla UCB and requires more memories. The computational complexity at step t is $\widetilde{\mathcal{O}}(Bt) \le \widetilde{\mathcal{O}}(BT)$. Comparing with vanilla UCB, the extra Bt is due to resampling. We also derive Theorem 0.1 below for MC quantile approximation error that provides us a theoretical guidance for the selection of B. In practice, the choice of B is seldom treated as a tuning parameter, but usually determined by the available computational resource. To reduce the computational cost, we could use the idea of Bag of Little Bootstraps [2]. For δ, from (2.5), a smaller value of δ enables us to calculate the quantile at a closer level to α but will result in a larger correction term. Since the correction term converges to 0 faster, we suggest a smaller δ. In Section 4.2, the $\delta = 1/(1+t)$ is essentially the confidence level, rather than a hyper-parameter δ in (2.5). The choice of $1/T^2$ led to an easy analysis. Using similar techniques in Chapter 8.2 of [1], we can derive a similar regret bound by setting $\alpha_t = 1/(t\log^{\alpha}(t))$ for any $\alpha > 0$. We re-run the experiments and there is no significant change.

Proof Clarifications. Thank you for pointing out the typos and notations! We have revised them accordingly. For Equation B.2, by the symmetric assumption of the reward, the distribution of $y_i - \mu$ is *exactly the same* as the distribution of $w_i(y_i - \mu)$ for Rademacher r.v. $\{w_i\}$. This implies $\mathbb{P}_{\boldsymbol{y}}(\frac{1}{n}\sum_{i=1}^{n}(y_i - \mu) > q_\alpha(\boldsymbol{y}_n - \mu)) = \mathbb{P}_{\boldsymbol{y},\boldsymbol{w}}(\frac{1}{n}\sum_{i=1}^{n} w_i(y_i - \mu) > q_\alpha((\boldsymbol{y}_n - \mu) \circ \boldsymbol{w}_n)) = \mathbb{E}_{\boldsymbol{w}}\mathbb{P}_{\boldsymbol{y}}\left(\frac{1}{n}\sum_{i=1}^{n} w_i(y_i - \mu) > q_\alpha((\boldsymbol{y}_n - \mu) \circ \boldsymbol{w}_n)\right)$. For the second one, instead of using conditional event, we could use union event trick: $\mathbb{P}(A) = \mathbb{P}(A \cap B) + \mathbb{P}(A \cap B^c) \le \mathbb{P}(A \cap B) + \mathbb{P}(B^c)$. By choosing $A = \{\bar{y}_n - \mu > q_{\alpha(1-\delta)}(\boldsymbol{y}_n - \bar{y}_n) + (2\log(2/\alpha\delta)/n)^{1/2}\varphi(\boldsymbol{y}_n)\}$ and $B = \{\boldsymbol{y}_n \in \mathcal{E}\}$, we can reach the conclusion.

Response to Reviewer 2

(1)(2). We will explicitly clarify and mention them in the beginning. (3). Yes, you are right. (4). We can derive a similar regret bound by setting $\alpha_t = 1/(t\log^{\alpha}(t))$ for any $\alpha > 0$. Please see our response for complexity part above. (5). We will change the argument to heavier tail than sub-Gaussian/exponential. (6). Thanks for the detailed references. We will add them in the introduction.

Response to Reviewer 3

Thank you for pointing the MC quantile approximation. We have derived the corresponding theorem for the control of the approximation of the bootstrapped quantile.

Theorem 0.1 (Monte Carlo Quantile Approximation)**.** Suppose the same conditions in Theorem 2.2 hold. We have $\mathbb{P}_{\boldsymbol{y},\boldsymbol{w}}(\bar{y}_n - \mu > \widetilde{q}_\alpha^B + \sqrt{\log(2/\alpha\delta)/n}\varphi(\boldsymbol{y}_n)) \le \alpha + \frac{\lfloor B\alpha \rfloor + 1}{B+1} \le 2\alpha + \frac{1}{B+1}$, where $\widetilde{q}_\alpha^B$ is the Monte Carlo approximated quantile defined in (D.1).

By replacing the true quantile q_α by a MC quantile $\widetilde{q}_\alpha^B$ based on B i.i.d bootstrapped weights, we lose at most $1/(B+1)$ for the confidence level. The proof is similar to Theorem 2.2 except for the control of i.i.d approximation error.

[1]. Bandit Algorithms. Cambridge University Press (2019). [2]. The Big Data Bootstrap. ICML (2012).