An Asymptotically Optimal Batched Algorithm for the Dueling Bandit Problem

Arpit Agarwal
Columbia University
arpit.agarwal@columbia.edu

Rohan Ghuge
University of Michigan
rghuge@umich.edu

Viswanath Nagarajan
University of Michigan
viswa@umich.edu

Abstract

We study the K-armed dueling bandit problem, a variation of the traditional multi-armed bandit problem in which feedback is obtained in the form of pairwise comparisons. Previous learning algorithms have focused on the *fully adaptive* setting, where the algorithm can make updates after every comparison. The "batched" dueling bandit problem is motivated by large-scale applications like web search ranking and recommendation systems, where performing sequential updates may be infeasible. In this work, we ask: *is there a solution using only a few adaptive rounds that matches the asymptotic regret bounds of the best sequential algorithms for K-armed dueling bandits?* We answer this in the affirmative *under the Condorcet condition*, a standard setting of the K-armed dueling bandit problem. We obtain asymptotic regret of $O(K^2 \log^2(K)) + O(K \log(T))$ in $O(\log(T))$ rounds, where T is the time horizon. Our regret bounds nearly match the best regret bounds known in the fully sequential setting under the Condorcet condition. Finally, in computational experiments over a variety of real-world datasets, we observe that our algorithm using $O(\log(T))$ rounds achieves almost the same performance as fully sequential algorithms (that use T rounds).

1 Introduction

The *K-armed dueling bandit* problem is a variation of the traditional multi-armed bandit problem in which feedback is obtained in the form of pairwise preferences. This problem has applications in a wide-variety of domains like search ranking, recommendation systems and sports ranking where eliciting qualitative feedback is easy while real-valued feedback is not easily interpretable; thus, it has been a popular topic of research in the machine learning community (see, for example, [51, 49, 47, 5, 54, 52, 53, 21, 32, 35, 36, 42, 18]).

Previous learning algorithms have focused on a *fully adaptive* setting; that is, the learning algorithm can make updates in a sequential fashion. Such updates might be impractical in large systems; for example, consider web-search ranking where the goal is to provide a list (usually *ranked*) of candidate documents to the user of the system in response to a query [41, 33, 50, 31]. Modern day search engines use hundred of parameters to compute a ranked list in response to a query, and online learning frameworks (based on user feedback) have been invaluable in automatically tuning these parameters [38]. However, given the scale of the system, it may be infeasible to adapt after each interaction: users may make multiple queries in a short time or multiple users may simultaneously query the system. Hence, we prefer solutions with *limited rounds* of adaptivity. Concretely, we ask: *is there a solution using only a few adaptive rounds that matches the asymptotic regret bounds of the best sequential algorithms for K-armed dueling bandits?*

This "batched" dueling bandit problem was introduced recently in [2]. Here, the learning algorithm's actions are partitioned into a limited number of rounds. In each round/batch, the algorithm commits to a *fixed set* of pairwise comparisons, and the feedback for all these comparisons is received

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

simultaneously. Then, the algorithm uses the feedback from the current batch of comparisons to choose comparisons for the next batch. [2] studied this problem under two different conditions: (i) the strong stochastic transitivity and stochastic triangle inequality (SST+STI) condition, which enforces a certain linear ordering over the arms; (ii) the Condorcet condition, which requires one arm to be superior to all others. Under SST+STI, their work provided almost tight upper and lower bounds on the trade-off between number of rounds and regret; in particular, they showed that one can achieve worst-case regret of $O(K \log^2 T)$ using $\Theta(\log T)$ rounds (T is the time-horizon).[1] Under the Condorcet condition, which is more general than SST+STI, they achieved a regret upper bound of $O(K^2 \log T)$ in $O(\log T)$ rounds. Previous work [54, 35] on fully sequential algorithms has shown that it is possible to achieve an *asymptotic* upper bound of $O(K^2 + K \log T)$ under the Condorcet condition. Very recently, [43] improved the sequential regret bound even further by obtaining regret $O(K \log T)$, which is the best possible even in the special case of SST+STI [49]. In the batched setting, the upper bound of [2] does not achieve this asymptotic optimality, irrespective of the number of batches, due to the presence of a K^2 *multiplicative* factor in the regret bound. Their work left open the possibility of obtaining a batched algorithm achieving asymptotic optimality under the Condorcet condition. In this paper, we nearly resolve this question, by providing an algorithm with $O(K^2 \log^2 K + K \log T)$ regret in $\Theta(\log T)$ rounds, under the Condorcet condition.

1.1 Contributions

- We design an algorithm, denoted `C2B`, for the batched dueling bandit problem, and analyze its regret under the Condorcet condition. This algorithm achieves a smooth trade-off between the expected regret and the number of batches, B.
- Crucially, when $B = \log(T)$, our regret bounds nearly match the best regret bounds [35, 54] known in the fully sequential setting. Hence, our results show that $O(\log T)$ rounds are sufficient to achieve asymptotically optimal regret as a function of T.
- Our results rely on new ideas for showing that the Condorcet winner arm can be 'trapped' using few adaptive rounds with high (constant) probability while incurring a reasonable amount of regret. We can then integrate over this space of probabilities to obtain a bound on the expected regret (in the same vein as [54]). Once the Condorcet arm is 'trapped', we can quickly eliminate all other 'sub-optimal' arms and minimize regret in the process.
- Finally, we run computational experiments to validate our theoretical results. We show that `C2B`, using $O(\log T)$ batches, achieves almost the same performance as fully sequential algorithms (which effectively use T batches) over a variety of real datasets.

1.2 Preliminaries

The K*-armed dueling bandit* problem [49] is an online optimization problem, where the goal is to find the best among K bandits $\mathcal{B} = \{1, \ldots, K\}$ using noisy pairwise comparisons with low *regret*. In each time-step, a noisy comparison between two arms (possibly the same), say (i, j), is performed. The outcome of the comparison is an independent random variable, and the probability of picking i over j is denoted $p_{i,j} = \frac{1}{2} + \Delta_{i,j}$ where $\Delta_{i,j} \in (-\frac{1}{2}, \frac{1}{2})$. Here, $\Delta_{i,j}$ can be thought of as a measure of distinguishability between the two arms, and we use $i \succ j$ when $\Delta_{i,j} > 0$. We also refer to $\Delta_{i,j}$ as the *gap* between i and j.

This problem has been studied under various conditions on the pairwise probabilities $p_{i,j}$'s. One such condition is the *strong stochastic transitivity* and *stochastic triangle inequality* (SST+STI) condition where there exists an ordering over arms, denoted by $\succeq$, such that for every triple $i \succeq j \succeq k$, we have $\Delta_{i,k} \geq \max\{\Delta_{i,j}, \Delta_{j,k}\}$, and $\Delta_{i,k} \leq \Delta_{i,j} + \Delta_{j,k}$ [49, 51]. In this paper, we work under the well-studied Condorcet winner condition, which is much more general than the SST+STI condition [47, 54, 35]. We say that arm i is a *Condorcet winner* if, and only if, $p_{i,j} > \frac{1}{2}$ for all $j \in \mathcal{B} \setminus \{i\}$. The *Condorcet condition* means that there exists a Condorcet winner.

Throughout the paper, we let a^* refer to the Condorcet arm. To further simplify notation, we define $\Delta_j = \Delta_{a^*,j}$; that is, the gap between a^* and j. We define the *regret* per time-step as follows: suppose arms i_t and j_t are chosen in time-step t, then the regret $r(t) = \frac{\Delta_{i_t} + \Delta_{j_t}}{2}$. The cumulative regret up to time T is $R(T) = \sum_{t=1}^{T} r(t)$, where T is the time horizon, and it's assumed that $K \leq T$. The

[1] They also gave a more complicated algorithm with regret $O(K \log^2 K \log T)$ in $O(\log T + \log K \log\log K)$ rounds, under the SST+STI condition.

Table 1: An overview of our results. Note that $\widetilde{O}$ hides polylog(K) factors.

Setting	Regret
Fully Adaptive	$O\left(\frac{K^2}{\Delta_{\min}^2}\right) + \sum_{j:\Delta_j>0} O\left(\frac{\log T}{\Delta_j}\right)$ [54, 35]
T rounds	$\sum_{j:\Delta_j>0} O\left(\frac{\log(T)}{\Delta_j}\right)$ [43]
Batched	$\sum_{j:\Delta_j>0} O\left(KT^{1/B} \cdot \frac{\log(T)}{\Delta_j}\right)$ [2]
B rounds	
Our Algorithm	
B rounds	$\widetilde{O}\left(T^{\frac{1}{B}} \cdot \frac{K^2}{\Delta_{\min}^2}\right) + O\left(T^{\frac{2}{B}} \cdot K^2\right) + \sum_j O\left(T^{\frac{1}{B}} \cdot \frac{\log(T)}{\Delta_j}\right)$
$\log(T)$ rounds	$\widetilde{O}\left(\frac{K^2}{\Delta_{\min}^2}\right) + \sum_j O\left(\frac{\log(T)}{\Delta_j}\right)$

cumulative regret can be equivalently stated as $R(T) = \frac{1}{2}\sum_{j=1}^{K} T_j \Delta_j$, where T_j denotes the number comparisons involving arm j. The goal of an algorithm is to minimize the cumulative $R(T)$. We define $\Delta_{\min} = \min_{j:\Delta_j>0} \Delta_j$ to be the smallest non-zero gap of any arm with a^*.

1.3 Batch Policies

In traditional bandit settings, actions are performed *sequentially*, utilizing the results of *all prior actions* in determining the next action. In the batched setting, the algorithm must commit to a round (or *batch*) of actions to be performed *in parallel*, and can only observe the results after all actions in the batch have been performed. More formally, given a number B of batches, the algorithm proceeds as follows. In each batch $r = 1, 2, \ldots B$, the algorithm first decides on the comparisons to be performed; then, *all* outcomes of the batch-r comparisons are received simultaneously. The algorithm can then, *adaptively*, select the next batch of comparisons. Note that even the size of the next batch can be adaptively decided based on the observations in previous batches. Finally, the total number of comparisons (across all batches) must sum to T. We assume that the values of T and B are known. Observe that when $T = B$, we recover the fully sequential setting.

1.4 Results and Techniques

We provide a overview of our results and prior results in Table 1. Given any integer $B \geq 1$, we obtain a B-round algorithm for the dueling bandit problem. We provide both high-probability and expected regret bounds, stated in the following theorems.

Theorem 1.1. *For any integer $B \geq 1$, there is an algorithm for the K-armed dueling bandit problem that uses at most B rounds with the following guarantee. For any $\delta > 0$, with probability at least $1 - \delta - \frac{1}{T}$, its regret under the Condorcet condition is at most*

$$R(T) \leq O\left(T^{1/B} \cdot \frac{K^2 \log(K)}{\Delta_{\min}^2} \cdot \log\left(\frac{\log K}{\Delta_{\min}}\right)\right) + O\left(T^{2/B} \cdot K^2 \cdot \sqrt{\frac{1}{\delta}}\right) + \sum_{j \neq a^*} O\left(\frac{T^{1/B} \cdot \log(KT)}{\Delta_j}\right).$$

Theorem 1.2. *For any integer $B \geq 1$, there is an algorithm for the K-armed dueling bandit problem that uses at most B rounds, with expected regret under the Condorcet condition at most*

$$\mathbb{E}[R(T)] = O\left(T^{1/B} \cdot \frac{K^2 \log(K)}{\Delta_{\min}^2} \cdot \log\left(\frac{\log K}{\Delta_{\min}}\right)\right) + O\left(T^{2/B} \cdot K^2\right) + \sum_{j \neq a^*} O\left(\frac{T^{1/B} \cdot \log(KT)}{\Delta_j}\right).$$

When the number of rounds $B = \log(T)$, we obtain a batched algorithm that achieves the asymptotic optimality (in terms of T), even for sequential algorithms. We formalize this observation in the following corollary.

Corollary 1.3. *There is an algorithm for the K-armed dueling bandit problem that uses at most $\log(T)$ rounds, with expected regret under the Condorcet condition at most*

$$\mathbb{E}[R(T)] = O\left(\frac{K^2\log(K)}{\Delta_{\min}^2} \cdot \log\left(\frac{\log K}{\Delta_{\min}}\right)\right) + \sum_{j \neq a^*} O\left(\frac{\log(KT)}{\Delta_j}\right).$$

By a lower-bound result from [2], it follows that no algorithm can achieve $O(\frac{K}{\Delta_{\min}} \cdot poly(\log T))$ regret using $o(\frac{\log T}{\log\log T})$ rounds, even under the SST+STI condition. So, the $O(\log T)$ rounds required to achieve asymptotic optimality in Corollary 1.3 is nearly the best possible.

Technical Challenges. The only prior approach for batched dueling bandits (under the Condorcet condition) is the algorithm PCOMP from [2], which performs all-pairs comparisons among arms in an active set. Such an approach cannot achieve regret better than $O(K^2 \log T)$ because the active set may remain large throughout. In order to achieve better regret bounds, [2] focus on the stronger SST+STI condition. In this setting, their main idea is to first sample a *seed set*, and use this seed set to eliminate sub-optimal arms. Their algorithm proceeds by performing all pairwise comparisons between the seed set and the set of active arms. However, the analysis of these 'seeded comparison' algorithms crucially rely on the total-ordering imposed by the SST and STI assumptions. Unfortunately, there is no such structure to exploit in the Condorcet setting: if the seed set does not contain the Condorcet winner, we immediately incur high regret.

The existing fully sequential algorithms such as RUCB [54] and RMED [35] are *highly adaptive* in nature. For instance, RUCB plays each *candidate* arm against an optimistic *competitor* arm using upper confidence bounds (UCB) on pairwise probabilities. This allows RUCB to quickly filter out candidates and uncover the Condorcet arm. Similarly, RMED plays each arm against a carefully selected *competitor* arm that is likely to beat this arm. However, such *competitors* can change frequently over trials in both RUCB and RMED. Since the batched setting requires comparisons to be predetermined, we do not have the flexibility to adapt to such changes in *competitors*. Hence, these existing fully sequential algorithms cannot be easily implemented in our setting.

Furthermore, we might also be tempted to consider an *explore-then-exploit* strategy where we first *explore* to find the Condorcet arm and *exploit* by playing this arm for remaining trials. However, this strategy is likely to fail because identifying the Condorcet arm with high probability might involve performing many comparisons, directly leading to high ($\Omega(K^2 \log T)$) regret; on the other hand, if the Condorcet winner is not identified with high probability, the exploit phase becomes expensive. This motivated us to consider algorithms that allow some form of *recourse*; that is, unless an arm is found to be sub-optimal, it must be given the opportunity to participate in the comparisons (as it could be the Condorcet winner).

The idea behind our algorithm is to identify the Condorcet winner a^* in a small *expected* number of rounds, after which it uses this arm as an "anchor" to eliminate sub-optimal arms while incurring low regret. To identify the best arm, in each round we define a candidate arm and compare it against arms that it "defeats". Arms that are not defeated by the candidate arm are compared to *all* active arms: this step ensures that the Condorcet winner is eventually discovered. We show that a^* becomes the candidate, and *defeats all other arms* within a small number of rounds (though the algorithm may not know if this has occurred). Additionally, once this condition is established, it remains invariant in future rounds. This allows us to eliminate sub-optimal arms and achieve low regret.

Comparison to `RUCB`. Initially, `RUCB` puts all arms in a pool of potential champions, and "optimistically" (using a upper confidence bound) performs all pairwise comparisons. Using these, it constructs a set of candidates C. If $|C|= 1$, then that arm is the hypothesised Condorcet winner and placed in a set B. Then, a randomized strategy is employed to choose a champion arm a_c (from sets C and B) which is compared to arm a_d which is most likely to beat it. The pair (a_c, a_d) is compared, the probabilities are updated and the algorithm continues. Although our algorithm also seeks to identify the best arm, we do not employ the UCB approach nor do we use any randomness. In their analysis, [54] show that the best arm eventually enters the set B, and remains in B: we also show a similar property for our algorithm in the analysis. Finally, similar to the analysis of [54], we first give a

high-probability regret bound for our algorithm which we then convert to a bound on the expected regret.

2 Related Work

The K-armed dueling bandit problem has been widely studied in recent years (we refer the reader to [46] for a comprehensive survey). Here, we survey the works that are most closely related to our setting. This problem was first studied in [49] under the SST and STI setting. The authors obtained a worst-case regret upper bound of $\widetilde{O}(K \log T/\Delta_{\min})$ and provided a matching lower bound. [51] considered a slightly more general version of the SST and STI setting and achieved an instance-wise optimal regret upper bound of $\sum_{j:\Delta_j>0} O\left(\log(T)/\Delta_j\right)$. Since, the SST+STI condition imposes a total order over the arms and might not hold for real-world datasets, [47] initiated the study of dueling bandits under the Condorcet winner condition. [47] proved a $O(K^2 \log T/\Delta_{\min})$ regret upper bound under the Condorcet condition, which was improved by [54] to $O(K^2/\Delta_{\min}^2) + \sum_{j:\Delta_j>0} O(\log T/\Delta_j^2)$. [35] achieved a similar but tighter KL divergence-based bound, which is shown to be *asymptotically instance-wise optimal* (even in terms constant factors). There are also other works that improve the dependence on K in the upper bound, but suffer a worse dependence on Δ_j's [53]. This problem has also been studied under other noise models such as utility based models [5] and other notions of regret [18]. Alternate notions of winners such as Borda winner [32], Copeland winner [52, 36, 48], and von Neumann winner [21] have also been considered. There are also several works on extensions of dueling bandits that allow multiple arms to be compared at once [45, 3, 44].

All of the aforementioned works on the dueling bandits problem are limited to the sequential setting. Recently, [2] initiated the study of the batched version of the K-armed dueling bandits. Their main results are under the SST and STI setting. They give two algorithms, called SCOMP and SCOMP2, for the batched K-armed dueling bandit problem. For any integer B, SCOMP uses at most $B+1$ batches and has expected regret bounded by $\sum_{j:\Delta_j>0} O(\sqrt{K}T^{1/B}\log(T)/\Delta_j)$. When $B = \log(T)$, this nearly matches (up to a factor of $\sqrt{K}$) the best known instance-dependent regret bound of $\sum_{j:\Delta_j>0} O(\log(T)/\Delta_j)$ obtained by [49]. SCOMP2 aims to achieve better worst-case regret: it uses at most $2B+1$ batches, and has regret $O\left(KBT^{1/B}\log(T)/\Delta_{\min}\right)$. Thus, when $B = \log(T)$, the expected worst-case regret is $O\left(K\log^2(T)/\Delta_{\min}\right)$, matching the best known result in the sequential setting up to an additional logarithmic factor. Under the Condorcet condition, [2] give a straightforward pairwise comparison algorithm (PCOMP), that achieves expected regret bounded by $O(K^2\log(T)/\Delta_{\min})$ in $\log(T)$ batches. They also provide a nearly matching lower bound of $\Omega(\frac{KT^{1/B}}{B^2\Delta_{\min}})$ for any B-batched algorithm. This implies that our bound (for B-round algorithms) in Theorem 1.2 cannot be significantly improved.

Recently, [43] designed a fully adaptive algorithm achieving an optimal regret of $\sum_{j:\Delta_j>0} \frac{O(\log T)}{\Delta_j}$ for dueling bandits under the Condorcet setting. This algorithm is based on the idea of *dueling* two classical bandit (MAB) algorithms against each other in a repeated zero-sum game with carefully designed rewards. The reward for one algorithm depends on the actions of the other; hence, these algorithms need to achieve *best-of-both-worlds* guarantee for both stochastic and adversarial settings. However, the approach of [43] is not directly applicable to the *batched* setting that we consider. This is because, as shown by [23], any B-round algorithm for batched MAB in the adversarial setting has regret $\Omega(T/B)$.

There has also been substantial work on best-arm or top-k identification using pairwise comparisons with limited adaptivity. [15, 14, 19] considered this problem under the noisy pairwise comparison setting, which is a special case of SST+STI. They showed that constant number of rounds of adaptivity are sufficient to solve these problem with the optimal sample complexity. [4] showed that one can also solve this problem under SST in constant number of rounds with the optimal sample complexity. However, these existing results focus on the SST setting, whereas we focus on the more general Condorcet winner setting. Moreover, these existing works focus on sample complexity for best-arm identification whereas our goal is regret minimization.

3 The Batched Algorithm

In this section, we describe a B-round algorithm for the K-armed dueling bandit problem under the Condorcet condition. Recall that given a set of K arms, $\mathcal{B} = \{1, \ldots, K\}$, and a positive integer

$B \leq \log(T)$, we wish to find a sequence of B batches of noisy comparisons with low regret. Given arms i and j, recall that $p_{i,j} = \frac{1}{2} + \Delta_{i,j}$ denotes the probability of i winning over j where $\Delta_{i,j} \in (-1/2, 1/2)$. We use a^* to denote the Condorcet winner; recall that a^* is a Condorcet winner if $p_{a^*,j} \geq 1/2$ for all $j \in \mathcal{B}$. To simplify notation, we use $\Delta_j = \Delta_{a^*,j}$. Before describing our algorithm, we first define some notation. We use $\mathcal{A}$ to denote the current set of *active* arms; i.e., the arms that have not been eliminated. We will use index r for rounds or batches. If pair (i, j) is compared in round r, it is compared $q_r = \lfloor q^r \rfloor$ times where $q = T^{1/B}$. We define the following quantities at the *end* of each round r:

- $N_{i,j}(r)$ is the total number of times the pair (i, j) has been compared.
- $\widehat{p}_{i,j}(r)$ is the frequentist estimate of $p_{i,j}$, i.e.,

$$\widehat{p}_{i,j}(r) = \frac{\#\ i \text{ wins against } j \text{ until end of round } r}{N_{i,j}(r)}. \tag{1}$$

- Two confidence-interval radii for each (i, j) pair:

$$c_{i,j}(r) = \sqrt{\frac{2\log(2K^2 q_r)}{N_{i,j}(r)}} \qquad \text{and} \qquad \gamma_{i,j}(r) = \sqrt{\frac{\log(K^2 BT)}{2N_{i,j}(r)}} \tag{2}$$

We now describe our B-round algorithm, called **C**ATCHING THE **C**ONDORCET WINNER IN **B**ATCHES (or, `C2B`). At a high-level, the algorithm identifies the best arm a^* in a small expected number of rounds, after which it uses this arm as an "anchor" to eliminate sub-optimal arms while incurring low regret. In every round r, we do the following:

1. We define a *defeated set* $D_r(i)$ for every active arm i; this set comprises arms that are defeated *with confidence* by i. Specifically, $j \in D_r(i)$ if $\widehat{p}_{i,j}(r-1) > 1/2 + c_{i,j}(r-1)$.
2. Then, we define a *candidate* i_r as the arm that defeats the most number of arms; that is, $i_r = \arg\max_{i \in \mathcal{A}} |D_r(i)|$.
3. For every arm $i \neq i_r$:
   - If $i \in D_r(i_r)$, then we compare i to i_r for q_r times. The idea here is to use i_r as an anchor against i. We will show that a^* becomes the candidate i_r in a small number of rounds. Then, this step ensures that we eliminate arms efficiently using a^* as an anchor.
   - If $i \notin D_r(i_r)$, then i is compared to all arms in $\mathcal{A}$ for q_r times. This step crucially protects the algorithm against cases where a sub-optimal arm becomes the candidate (and continues to become the candidate). For example, suppose $K = [5]$ and the arms are linearly ordered as $1 \succ 2 \succ \cdots \succ 5$. Furthermore suppose that in some round r, we have that (a) 2 defeats $3, 4, 5$ and (b) 1 (best arm) defeats 2 but not the others. So, 2 is the candidate in round r; if 1 is not compared to $3, 4, 5$, then 2 would continue to be the candidate (leading to high regret).
4. If, for any arm j, there is arm i such that $\widehat{p}_{i,j}(r) > \frac{1}{2} + \gamma_{i,j}(r)$, then j is eliminated from $\mathcal{A}$.

This continues until T total comparisons are performed. See Algorithm 1 for a formal description. The main result of this section is to show that `C2B` achieves the guarantees stated in Theorems 1.1 and 1.2.

Overview of the Analysis. We provide a brief outline of the proofs of our main results. Towards proving Theorem 1.1, we first define two events:

- The first event, denoted G, ensures that a^* is not eliminated during the execution of `C2B`. We show that $\mathbf{P}(G) \geq 1 - 1/T$.
- The second event, denoted $E(\delta)$, says that there exists a round $C(\delta)$ (defined later) such that for all $r > C(\delta)$, the estimate $\widehat{p}_{i,j}(r-1)$ satisfies the confidence interval of $c_{i,j}(r-1)$. Moreover, $\mathbf{P}(E(\delta)) \geq 1 - \delta$.

By union bound, $\mathbf{P}(G \cap E(\delta)) \geq 1 - \delta - 1/T$. Together, we use G and $E(\delta)$ to argue that:

Algorithm 1 C2B (CATCHING THE CONDORCET WINNER IN BATCHES)

Input: Arms $\mathcal{B}$, time-horizon T, integer $B \geq 1$
active arms $\mathcal{A} \leftarrow \mathcal{B}$, $r \leftarrow 1$, emprical probabilities $\widehat{p}_{i,j}(0) = \frac{1}{2}$ for all $i, j \in \mathcal{B}^2$
while number of comparisons $\leq T$ **do**
  if $\mathcal{A} = \{i\}$ for some i **then** play (i, i) for remaining trials
  $D_r(i) \leftarrow \{j \in \mathcal{A} : \widehat{p}_{i,j}(r-1) > \frac{1}{2} + c_{i,j}(r-1)\}$
  $i_r \leftarrow \arg\max_{i \in \mathcal{A}} |D_r(i)|$
  for $i \in \mathcal{A} \setminus \{i_r\}$ **do**
    if $i \in D_r(i_r)$ **then**
      compare (i_r, i) for q_r times
    else
      for each $j \in \mathcal{A}$, compare (i, j) for q_r times
  compute $\widehat{p}_{i,j}(r)$ values
  if $\exists i, j : \widehat{p}_{i,j}(r) > \frac{1}{2} + \gamma_{i,j}(r)$ **then**
    $\mathcal{A} \leftarrow \mathcal{A} \setminus \{j\}$
  $r \leftarrow r + 1$

- the best arm, a^*, is *not defeated* by any arm i in any round $r > C(\delta)$ (formalized in Lemma 3.5),
- and that there exists a round $r(\delta) \geq C(\delta)$ such that for every round after $r(\delta)$, arm a^* defeats *every other arm* (formalized in Lemma 3.7).

Intuitively, these observations imply that our algorithm identifies the best arm after $r(\delta)$ rounds. Thus, beyond round $r(\delta)$, we only perform pairwise comparisons of the form (a^*, i) for $i \neq a^*$: thus, a^* is used as an anchor to eliminate sub-optimal arms. We then analyze the regret in two parts: (i) regret incurred up to round $r(\delta)$, which is upper bounded by $K^2 \sum_{r \leq r(\delta)} q^r$ and (ii) regret after $r(\delta)$, which is the regret incurred in eliminating sub-optimal arms using a^* as an anchor. Finally, we can use the high-probability bound to also obtain a bound on the expected regret, proving Theorem 1.2. We provide some details of the proof of Theorem 1.1. We defer the proof of Theorem 1.2 to Appendix D.

3.1 The Analysis

In this section, we give high-probability and expected regret bounds for C2B. Recall that $q = T^{1/B}$, and that $q \geq 2$. The following lemma is used to prove that a^* is never eliminated. We defer the proofs of the Lemmas 3.1, 3.2 to Appendix B.

Lemma 3.1. *For any batch $r \in [B]$, and for any pair (i, j), we have*

$$\mathbf{P}\left(|\widehat{p}_{i,j}(r) - p_{i,j}| > \gamma_{i,j}(r)\right) \leq 2\eta,$$

where $\eta = 1/K^2BT$.

We first define the *good* event G as follows.

Definition 3.1 (Event G)**.** *An estimate $\widehat{p}_{i,j}(r)$ at the end of batch r is **strongly-correct** if $|\widehat{p}_{i,j}(r) - p_{i,j}| \leq \gamma_{i,j}(r)$. We say that event G occurs if every estimate in every batch $r \in [B]$ is strongly-correct.*

The following two lemmas show that G occurs with high probability and that a^* is not eliminated under G.

Lemma 3.2. *The probability that every estimate in every batch of* C2B *is strongly-correct is at least $1 - 1/T$.*

Lemma 3.3. *Conditioned on G, a^* is never eliminated from $\mathcal{A}$ in the elimination step of* C2B.

Proof. In C2B, an arm j is deleted in batch r iff there is an arm $i \in \mathcal{A}$ with $\widehat{p}_{i,j}(r) > \frac{1}{2} + \gamma_{i,j}(r)$. If a^* is eliminated due to some arm j, then by definition of event G, we get $p_{j,a^*} \geq \widehat{p}_{i,j}(r) - \gamma_{i,j}(r) > \frac{1}{2}$, a contradiction. □

3.1.1 High-probability Regret Bound

In this section, we give details required to prove Theorem 1.1. Fix any $\delta > 0$. We first define another good event as follows.

Definition 3.2 (Event $E(\delta)$)**.** *An estimate $\widehat{p}_{i,j}(r)$ in batch r is **weakly-correct** if $|\widehat{p}_{i,j}(r) - p_{i,j}| \leq c_{i,j}(r)$. Let $C(\delta) := \lceil \frac{1}{2} \log_q(1/\delta) \rceil$. We say that event $E(\delta)$ occurs if for each batch $r \geq C(\delta)$, every estimate is weakly-correct.*

The next lemma shows that $E(\delta)$ occurs with probability at least $1 - \delta$.

Lemma 3.4. *For all $\delta > 0$, we have*

$$\mathbf{P}(\neg E(\delta)) \;=\; \mathbf{P}\left(\exists r \geq C(\delta), i, j : |\widehat{p}_{i,j}(r) - p_{i,j}| > c_{i,j}(r)\right) \;\leq\; \delta.$$

We will analyze our algorithm under both events G and $E(\delta)$. Note that event G is required to ensure that a^* is not eliminated in rounds before $C(\delta)$ (where the Lemma 3.4 does not apply).

Lemma 3.5. *Conditioned on G and $E(\delta)$, for any round $r > C(\delta)$, arm a^* is not defeated by any other arm, i.e., $a^* \notin \cup_{i \neq a^*} D_r(i)$.*

To proceed, we need the following definitions.

Definition 3.3. *The candidate i_r of round r is called the **champion** if $|D_r(i_r)| = |\mathcal{A}| - 1$; that is, if i_r defeats every other active arm.*

Definition 3.4. *Let $r(\delta) \geq C(\delta) + 1$ be the smallest integer such that*

$$q^{r(\delta)} \geq 2A \log A, \qquad \textit{where } A := \frac{32}{\Delta_{\min}^2} \cdot \log(2K^2).$$

We use the following inequality based on this choice of $r(\delta)$.

Lemma 3.6. *The above choice of $r(\delta)$ satisfies*

$$q^r > \frac{8}{\Delta_{\min}^2} \cdot \log\left(2K^2 q_r\right), \qquad \forall r \geq r(\delta).$$

Proof of Lemma 3.6. Using the fact that $q_r \leq q^r$, it suffices to show $q^r \geq \frac{8}{\Delta_{\min}^2} \cdot \big(\log(2K^2) + \log q^r\big)$. Moreover,

$$\log(2K^2) + \log q^r \leq \big(1 + \log(2K^2)\big) \cdot (1 + \log q^r) \leq 4 \cdot \log(2K^2) \cdot \log q^r,$$

where the last inequality uses $K \geq 2$, $r \geq r(\delta) \geq 1$ and $q \geq 2$. So, it suffices to show:

$$q^r > A \cdot \log(q^r), \quad \forall r \geq r(\delta), \qquad \text{where } A = \frac{32}{\Delta_{\min}^2} \cdot \log(2K^2) \tag{3}$$

Below, let $x = q^r$, $R := 2A \log A$ and function $f(x) := x - A \log x$. We will show that $f(x) > 0$ for all $x \geq R$, which would imply (3) because $q^{r(\delta)} \geq R$. As $R \geq A$, and f is increasing for $x \geq A$, it suffices to show that $f(R) \geq 0$. Indeed,

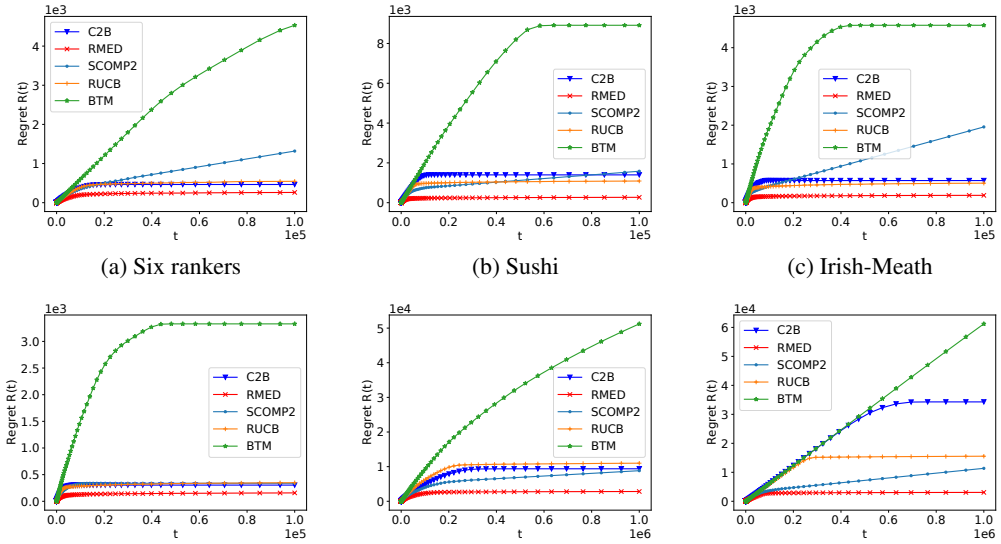
$$\frac{f(R)}{A} = 2 \log A - \log(2A \log A) = \log A - \log(2 \log A) > 0,$$

where the inequality uses $A \geq 8$. □

Then, we have the following.

Lemma 3.7. *Conditioned on G and $E(\delta)$, the best arm a^* is the champion in every round $r > r(\delta)$.*

We now have all components required to prove Theorem 1.1; its proof, and the proofs of the aforementioned lemmas can be found in Appendix C.



(a) Six rankers (b) Sushi (c) Irish-Meath

(d) Irish-Dublin (e) MSLR30 (f) Yahoo30

Figure 1: Regret v/s t plots of algorithms when $B = \lfloor \log(T) \rfloor$

4 Computational Results

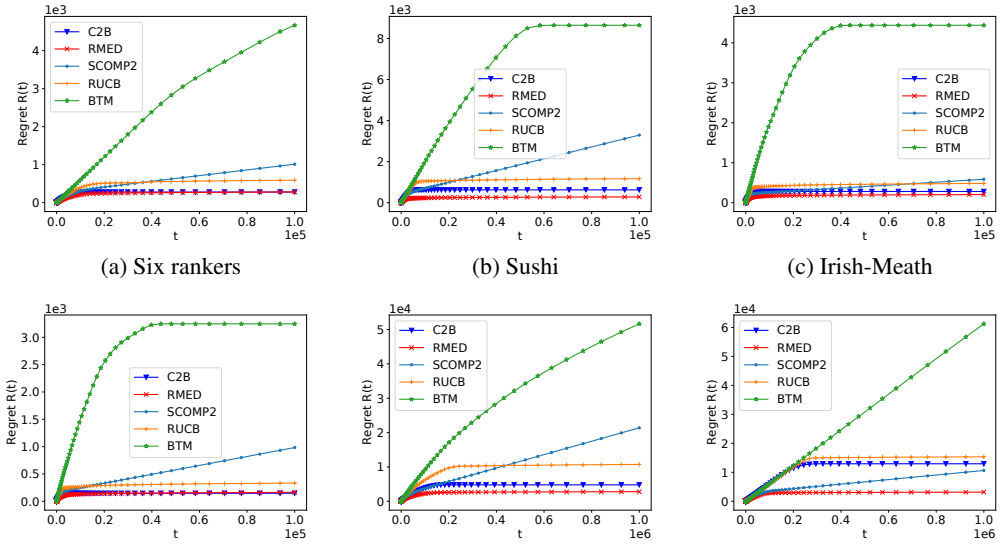
In this section, we provide details of our computational experiments. The goal of our experiments is to answer the following questions: (i) How does the regret of `C2B` using $B = \lfloor \log(T) \rfloor$ batches compare to that of existing fully sequential as well as batched algorithms? and (ii) Can the regret of `C2B` match the regret of the best known sequential algorithms; if yes, then how many rounds suffice to achieve this? Towards answering (i), we compare `C2B` to a representative set of sequential algorithms for dueling bandits using the library due to [35]. We compare `C2B` to the sequential algorithms RUCB [54], RMED [35], and BEAT-THE-MEAN (BTM) [51]. We allow these algorithms to work as prescribed; that is, they work in $B = T$ batches. The reason that we chose these sequential algorithms is that our batched algorithm (`C2B`) is based on a similar paradigm, and such a comparison demonstrates the power of adaptivity in this context. We also compare `C2B` to the batched algorithm SCOMP2 [2]. We plot the cumulative regret $R(t)$ incurred by the algorithms against time t. We set $B = \lfloor \log(T) \rfloor$ for `C2B` and SCOMP2 in this experiment. For (ii), we increased B by a small amount; we found that the performance of `C2B` improves noticeably when given a constant number of additional rounds (we use $B = \lfloor \log(T) \rfloor + 6$ in this experiment). We perform these experiments using the following real-world datasets.

Six rankers. This dataset is based on the 6 retrieval functions used in the engine of ArXiv.org.

Sushi. The Sushi dataset is based on the Sushi preference dataset [34] that contains the preference data regarding 100 types of Sushi. A preference dataset using the top-16 most popular types of sushi is obtained.

Irish election data. The Irish election data for Dublin and Meath is available at *preflib.org*. It contains partial preference orders over candidates. As in [3], these are transformed into preference matrices by selecting a subset of candiates to ensure that a Condorcet winner exists. There are 12 candidates in the **Irish-Meath** dataset, and 8 in the **Irish-Dublin** dataset.

MSLR and Yahoo! data. We also run experiments on two web search ranking datasets: the Microsoft Learning to Rank (MSLR) dataset [40] and the Yahoo! Learning to Rank Challenge Set 1 [16]. These datasets have been used in prior work on online ranker evaluation [53, 37]. We use preference matrices generated using the "navigational" configuration (see [37] for details). The MSLR dataset has 136 rankers and the Yahoo! dataset has 700 rankers. We sample 30 rankers from each dataset while ensuring the existence of a Condorcet winner. In this way, we obtain two datasets, denoted **MSLR30** and **Yahoo30**.



(a) Six rankers (b) Sushi (c) Irish-Meath

(d) Irish-Dublin (e) MSLR30 (f) Yahoo30

Figure 2: Regret v/s t plots of algorithms when $B = \lfloor \log(T) \rfloor + 6$

Note that there exists a Condorcet winner in all datasets. We repeat each experiment 20 times and report the average regret. In our algorithm, we use the *KL-divergence based confidence bound* due to [35] for elimination as it performs much better empirically, and our theoretical bounds continue to hold (see §E). This KL-divergence based elimination criterion eliminates an arm i in round r if $I_i(r) - I^*(r) > \log(T) + f(K)$ where $I_i(r) = \sum_{j:\widehat{p}_{i,j}(r) < \frac{1}{2}} N_{i,j}(r) \cdot D_{\mathrm{KL}}(\widehat{p}_{i,j}(r), \frac{1}{2})$ and $I^*(r) = \min_{j \in [K]} I_i(r)$.

Computational Results. As mentioned earlier, we compare our algorithms against a representative set of sequential dueling bandits algorithms (RUCB, RMED, and BTM). We set $\alpha = 0.51$ for RUCB, and $f(K) = 0.3K^{1.01}$ for RMED and `C2B`, and $\gamma = 1.3$ for BTM: these parameters are known to perform well both theoretically and empirically [35]. We set $T = 10^6$ for MSLR30 and Yahoo30 datasets (as they have larger number of arms), and $T = 10^5$ for the remaining four. For the first set of experiments, we set $B = \lfloor \log(T) \rfloor$. We observe that `C2B` always outperforms BTM and beats SCOMP2 on most of the datasets. We observe that even when SCOMP2 beats `C2B` it has a slightly linear curve (implying that its regret would keep increasing as T increases) while the regret curve of `C2B` is mostly flat. Furthermore, `C2B` performs comparably to RUCB in all datasets except Yahoo30. We plot the results in Figure 1. In the second set of experiments, we set $B = \lfloor \log(T) \rfloor + 6$. We observe that `C2B` always outperforms RUCB and, in fact, performs comparably to RMED on all datasets except Yahoo30. We plot the results in Figure 2. Finally, we note that SCOMP2 exhibits varying performance across runs (even on the same dataset) and we think that this is due to the randomness involved in selecting the "seed set".

5 Conclusion

In this paper, we proposed a batched algorithm, named `C2B`, for the K-armed dueling bandit problem. Assuming the existence of a Condorcet winner, we show both high-probability and expected regret bounds for `C2B` that trade-off smoothly with the number of batches. Furthermore, we obtain asymptotic regret of $O(K^2 \log^2(K)) + O(K \log(T))$ in $O(\log(T))$ batches, nearly matching the best regret bounds known in the fully sequential setting under the Condorcet condition. Our computational results show that `C2B`, using $O(\log(T))$ batches, achieves almost the same performance as fully sequential algorithms over a variety of real-world datasets. A direction for future research is to design batched algorithms for the K-armed dueling bandit problem when a Condorcet winner does not exist; for example, designing an algorithm for a more general concept of winner, such as *Copeland winner* [48] or *von Neumann winner* [22].

References

[1] A. Agarwal, S. Assadi, and S. Khanna. Stochastic submodular cover with limited adaptivity. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms*, page 323–342, 2019.

[2] A. Agarwal, R. Ghuge, and V. Nagarajan. Batched dueling bandits. In *Proceedings of the 39th International Conference on Machine Learning*, pages 89–110, 2022.

[3] A. Agarwal, N. Johnson, and S. Agarwal. Choice bandits. In *NeurIPS*, 2020.

[4] A. Agarwal, S. Khanna, and P. Patil. PAC top-k identification under SST in limited rounds. In G. Camps-Valls, F. J. R. Ruiz, and I. Valera, editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*, volume 151 of *Proceedings of Machine Learning Research*, pages 6814–6839. PMLR, 2022.

[5] N. Ailon, Z. Karnin, and T. Joachims. Reducing Dueling Bandits to Cardinal Bandits. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

[6] E. Balkanski, A. Breuer, and Y. Singer. Non-monotone submodular maximization in exponentially fewer iterations. In *Advances in Neural Information Processing Systems*, pages 2359–2370, 2018.

[7] E. Balkanski, A. Rubinstein, and Y. Singer. An exponential speedup in parallel running time for submodular maximization without loss in approximation. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 283–302, 2019.

[8] E. Balkanski and Y. Singer. The adaptive complexity of maximizing a submodular function. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1138–1151, 2018.

[9] E. Balkanski and Y. Singer. Approximation guarantees for adaptive sampling. In *Proceedings of the 35th International Conference on Machine Learning*, pages 393–402, 2018.

[10] N. Bansal, A. Gupta, J. Li, J. Mestre, V. Nagarajan, and A. Rudra. When LP is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica*, 63(4):733–762, 2012.

[11] N. Bansal and V. Nagarajan. On the adaptivity gap of stochastic orienteering. *Math. Program.*, 154(1-2):145–172, 2015.

[12] S. Behnezhad, M. Derakhshan, and M. Hajiaghayi. Stochastic matching with few queries: (1-ϵ) approximation. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1111–1124, 2020.

[13] A. Bhalgat, A. Goel, and S. Khanna. Improved approximation results for stochastic knapsack problems. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1647–1665, 2011.

[14] M. Braverman, J. Mao, and Y. Peres. Sorted top-k in rounds. In A. Beygelzimer and D. Hsu, editors, *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, volume 99 of *Proceedings of Machine Learning Research*, pages 342–382. PMLR, 2019.

[15] M. Braverman, J. Mao, and S. M. Weinberg. Parallel Algorithms for Select and Partition with Noisy Comparisons. In *Proceedings of the 48th Annual Symposium on the Theory of Computing*, 2016.

[16] O. Chapelle and Y. Chang. Yahoo! learning to rank challenge overview. In *Proceedings of the 2010 International Conference on Yahoo! Learning to Rank Challenge - Volume 14*, page 1–24. JMLR.org, 2010.

[17] C. Chekuri and K. Quanrud. Parallelizing greedy for submodular set function maximization in matroids and beyond. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 78–89, 2019.

[18] B. Chen and P. I. Frazier. Dueling Bandits with Weak Regret. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[19] V. Cohen-Addad, F. Mallmann-Trenn, and C. Mathieu. Instance-optimality in the noisy value-and comparison-model. In S. Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2124–2143. SIAM, 2020.

[20] B. C. Dean, M. X. Goemans, and J. Vondrák. Approximating the stochastic knapsack problem: The benefit of adaptivity. *Math. Oper. Res.*, 33(4):945–964, 2008.

[21] M. Dudik, K. Hofmann, R. E. Schapire, A. Slivkins, and M. Zoghi. Contextual Dueling Bandits. In *Proceedings of the 28th Conference on Learning Theory*, 2015.

[22] M. Dudík, K. Hofmann, R. E. Schapire, A. Slivkins, and M. Zoghi. Contextual dueling bandits. In P. Grünwald, E. Hazan, and S. Kale, editors, *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, volume 40 of *JMLR Workshop and Conference Proceedings*, pages 563–587. JMLR.org, 2015.

[23] H. Esfandiari, A. Karbasi, A. Mehrabian, and V. Mirrokni. Regret bounds for batched bandits. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8):7340–7348, May 2021.

[24] H. Esfandiari, A. Karbasi, and V. Mirrokni. Adaptivity in adaptive submodularity. In *Proceedings of 34th Conference on Learning Theory*, volume 134, pages 1823–1846. PMLR, 2021.

[25] Z. Gao, Y. Han, Z. Ren, and Z. Zhou. Batched multi-armed bandits problem. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 501–511, 2019.

[26] R. Ghuge, A. Gupta, and V. Nagarajan. The power of adaptivity for stochastic submodular cover. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3702–3712. PMLR, 18–24 Jul 2021.

[27] D. Golovin and A. Krause. Adaptive submodularity: A new approach to active learning and stochastic optimization. *CoRR*, abs/1003.3967, 2017.

[28] S. Guha and K. Munagala. Multi-armed bandits with metric switching costs. In *Automata, Languages and Programming, 36th Internatilonal Colloquium (ICALP)*, pages 496–507, 2009.

[29] A. Gupta, R. Krishnaswamy, V. Nagarajan, and R. Ravi. Running errands in time: Approximation algorithms for stochastic orienteering. *Math. Oper. Res.*, 40(1):56–79, 2015.

[30] A. Gupta and V. Nagarajan. A stochastic probing problem with applications. In *Integer Programming and Combinatorial Optimization - 16th International Conference*, pages 205–216, 2013.

[31] K. Hofmann, S. Whiteson, and M. Rijke. Balancing exploration and exploitation in listwise and pairwise online learning to rank for information retrieval. *Inf. Retr.*, 16(1):63–90, feb 2013.

[32] K. Jamieson, S. Katariya, A. Deshpande, and R. Nowak. Sparse Dueling Bandits. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, 2015.

[33] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 133–142, New York, NY, USA, 2002. Association for Computing Machinery.

[34] T. Kamishima. Nantonac collaborative filtering: recommendation based on order responses. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24 - 27, 2003*, pages 583–588, 2003.

[35] J. Komiyama, J. Honda, H. Kashima, and H. Nakagawa. Regret Lower Bound and Optimal Algorithm in Dueling Bandit Problem. In *Proceedings of the 28th Conference on Learning Theory*, 2015.

[36] J. Komiyama, J. Honda, and H. Nakagawa. Copeland Dueling Bandit Problem: Regret Lower Bound, Optimal Algorithm, and Computationally Efficient Algorithm. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

[37] C. Li, I. Markov, M. de Rijke, and M. Zoghi. Mergedts: A method for effective large-scale online ranker evaluation. *ACM Trans. Inf. Syst.*, 38(4):40:1–40:28, 2020.

[38] T.-Y. Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331, mar 2009.

[39] V. Perchet, P. Rigollet, S. Chassang, and E. Snowberg. Batched bandit problems. *The Annals of Statistics*, 44(2):660–681, 2016.

[40] T. Qin and T.-Y. Liu. Introducing letor 4.0 datasets. *ArXiv*, abs/1306.2597, 2013.

[41] F. Radlinski, M. Kurup, and T. Joachims. How does clickthrough data reflect retrieval quality? In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, page 43–52, New York, NY, USA, 2008. Association for Computing Machinery.

[42] S. Ramamohan, A. Rajkumar, and S. Agarwal. Dueling Bandits : Beyond Condorcet Winners to General Tournament Solutions. In *Advances in Neural Information Processing Systems 29*, 2016.

[43] A. Saha and P. Gaillard. Versatile dueling bandits: Best-of-both world analyses for learning from relative preferences. In *International Conference on Machine Learning*, pages 19011–19026. PMLR, 2022.

[44] A. Saha and A. Gopalan. Combinatorial bandits with relative feedback. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 983–993, 2019.

[45] Y. Sui, V. Zhuang, J. W. Burdick, and Y. Yue. Multi-dueling Bandits with Dependent Arms. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*, 2017.

[46] Y. Sui, M. Zoghi, K. Hofmann, and Y. Yue. Advancements in dueling bandits. In J. Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 5502–5510. ijcai.org, 2018.

[47] T. Urvoy, F. Clerot, R. Feraud, and S. Naamane. Generic Exploration and K-armed Voting Bandits. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

[48] H. Wu and X. Liu. Double thompson sampling for dueling bandits. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 649–657, 2016.

[49] Y. Yue, J. Broder, R. Kleinberg, and T. Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012. JCSS Special Issue: Cloud Computing 2011.

[50] Y. Yue and T. Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 1201–1208, New York, NY, USA, 2009. Association for Computing Machinery.

[51] Y. Yue and T. Joachims. Beat the mean bandit. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

[52] M. Zoghi, Z. Karnin, S. Whiteson, and M. de Rijke. Copeland Dueling Bandits. In *Advances in Neural Information Processing Systems 28*, 2015.

[53] M. Zoghi, S. Whiteson, and M. de Rijke. MergeRUCB: A method for large-scale online ranker evaluation. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining*, 2015.

[54] M. Zoghi, S. Whiteson, R. Munos, and M. de Rijke. Relative Upper Confidence Bound for the K-Armed Dueling Bandit Problem. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]

(b) Did you describe the limitations of your work? [Yes] We design a batched algorithm for the K-armed dueling bandit when the SST+STI condition does not hold; we work under the Condorcet condition. In §5, we pose designing algorithms for even more general settings as an open problem.

(c) Did you discuss any potential negative societal impacts of your work? [No] We don't forsee any negative societal impacts of our work.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

   (a) Did you state the full set of assumptions of all theoretical results? [Yes]

   (b) Did you include complete proofs of all theoretical results? [Yes]

3. If you ran experiments...

   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We include the code in the supplemental material.

   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We include the details in §4.

   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We include the details in §4.

   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We include the details in the appendix.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes]

   (b) Did you mention the license of the assets? [N/A]

   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] The library we used is publicly available.

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Further Related Work

Batched algorithms for multi-armed bandits. Batched processing for the stochastic multi-armed bandit problem has been investigated in the past few years. A special case when there are two bandits was studied by [39]. They obtain a worst-case regret bound of $O\left(\left(\frac{T}{\log(T)}\right)^{1/B}\frac{\log(T)}{\Delta_{\min}}\right)$. [25] studied the general problem and obtained a worst-case regret bound of $O\left(\frac{K\log(K)T^{1/B}\log(T)}{\Delta_{\min}}\right)$, which was later improved by [23] to $O\left(\frac{KT^{1/B}\log(T)}{\Delta_{\min}}\right)$. Furthermore, [23] obtained an instance-dependent regret bound of $\sum_{j:\Delta_j>0} T^{1/B}O\left(\frac{\log(T)}{\Delta_j}\right)$. Our results for batched dueling bandits are of a similar flavor; that is, we get a similar dependence on T and B. [23] also give batched algorithms for stochastic linear bandits and adversarial multi-armed bandits.

Adaptivity in combinatorial optimization. Adaptivity and batch processing has been recently studied for stochastic submodular cover [27, 1, 24, 26], and for various stochastic "maximization" problems such as knapsack [20, 13], matching [10, 12], probing [30] and orienteering [28, 29, 11]. Recently, there have also been several results examining the role of adaptivity in (deterministic) submodular optimization; e.g. [8, 6, 9, 7, 17].

B Missing Proofs from §3.1

Proof of Lemma 3.1. Note that $\mathbb{E}[\widehat{p}_{i,j}(r)] = p_{i,j}$, and applying Hoeffding's inequality gives

$$\mathbf{P}\left(|\widehat{p}_{i,j}^{r+1} - p_{i,j}| > \gamma_{i,j}(r)\right) \leq 2\exp\left(-2N_{i,j}(r)\cdot\gamma_{i,j}(r)^2\right) \leq 2\eta.$$

□

Proof of Lemma 3.2. Applying Lemma 3.1 and taking a union bound over all pairs and batches, we get that the probability that some estimate is incorrect is at most $\binom{K}{2}\times B\times 2\eta \leq \frac{1}{T}$ where $\eta = 1/K^2BT$. Thus, $\mathbf{P}(\overline{G}) \leq \frac{1}{T}$. □

Proof of Lemma 3.3. In C2B, an arm j is deleted in batch r iff there is an arm $i \in \mathcal{A}$ with $\widehat{p}_{i,j}(r) > \frac{1}{2} + 2\gamma_{i,j}(r)$. If a^* is eliminated due to some arm j, then by definition of event G, we get $p_{j,a^*} > \frac{1}{2} + \gamma_{i,j}(r) > \frac{1}{2}$, a contradiction. □

C Missing Proofs from §3.1.1

Proof of Lemma 3.4. For any pair i, j of arms and round r, let $B_{i,j}(r)$ denote the event that $|\widehat{p}_{i,j}(r) - p_{i,j}| > c_{i,j}(r)$. Note that $N_{ij}(r) \leq \sum_{s=1}^{r} q_s \leq 2q_r$. For any integer n, let $s_{ij}(n)$ denote the sample average of n independent Bernoulli r.v.s with probability p_{ij}. By Hoeffding's bound,

$$\mathbf{P}[|s_{ij}(n) - p_{ij}| > c] \leq 2e^{-2nc^2}, \qquad \text{for any } c \in [0,1].$$

We now bound

$$\begin{aligned}
\mathbf{P}[B_{ij}(r)] &\leq \sum_{n=0}^{2q_r}\mathbf{P}[B_{ij}(r) \wedge N_{ij}(r) = n] \\
&\leq \sum_{n=0}^{2q_r}\mathbf{P}\left[|s_{ij}(n) - p_{ij}| > \sqrt{\frac{2\log(2K^2q_r)}{n}}\right] \leq \sum_{n=0}^{2q_r} 2\exp\left(-2n\cdot\frac{2\log(2K^2q_r)}{n}\right) \\
&\leq 4q_r\cdot\frac{1}{(2K^2q_r)^4} \leq \frac{1}{4K^2\cdot q_r^2}
\end{aligned}$$

The second inequality uses the definition of $c_{ij}(r)$ when $N_{ij}(r) = n$. The last inequality uses $K \geq 2$. Next, by a union bound over arms and rounds, we can write the desired probability as

$$\begin{aligned}
\mathbf{P}(\exists r \geq C(\delta), i, j : B_{i,j}(r)) &\leq \sum_{r \geq C(\delta)} \sum_{i<j} \mathbf{P}(B_{i,j}(r)) \\
&\leq \sum_{r \geq C(\delta)} \binom{K}{2} \cdot \frac{1}{4K^2 \cdot q_r^2} \leq \sum_{r \geq C(\delta)} \frac{1}{8q_r^2} \\
&\leq \sum_{r \geq C(\delta)} \frac{1}{2q^{2r}} = \frac{1}{2q^{2C(\delta)}} \cdot \left(1 + \frac{1}{q^2} + \frac{1}{q^4} + \cdots\right) \leq \frac{1}{q^{2C(\delta)}} \leq \delta
\end{aligned} \tag{4}$$

The second inequality above uses the bound on $\mathbf{P}[B_{ij}(r)]$. The first inequality in (4) uses $q_r = \lfloor q^r \rfloor \geq q^r - 1 \geq \frac{q^r}{2}$ as $q \geq 2$. The last inequality in (4) uses the definition of $C(\delta)$.

The lemma now follows by the definition of event $\neg E(\delta)$ as $\exists r \geq C(\delta), i, j : B_{i,j}(r)$. □

Proof of Lemma 3.5. Fix any round $r \geq C(\delta) + 1$. Suppose that $a^* \in D_r(i)$ for some other arm i. This implies that $\widehat{p}_{i,a^*}(r-1) > \frac{1}{2} + c_{i,a^*}(r-1)$. But under event $E(\delta)$, we have $|\widehat{p}_{i,a^*}(r-1) - p_{i,a^*}| \leq c_{i,a^*}(r-1)$ because $r - 1 \geq C(\delta)$. Combined, these two observations imply $p_{i,a^*} > \frac{1}{2}$, a contradiction. □

Proof of Lemma 3.7. We first argue that a^* is compared to all active arms in each round $r \geq r(\delta)$. By Lemma 3.3, we know $a^* \in \mathcal{A}$. By Lemma 3.5, we have $a^* \notin D_r(j)$ for all $j \neq a^*$ because $r \geq r(\delta) \geq 1 + C(\delta)$. If candidate $i_r \neq a^*$, then a^* will be compared to all $j \in \mathcal{A}$ (since $a^* \notin D_r(i_r)$). On the other hand, if $i_r = a^*$, then (1) for any $j \in D_r(a^*)$, arm j is only compared to a^*, and (2) for any $j \in \mathcal{A} \setminus D_r(a^*)$, arm j is compared to all active arms including a^*.

Next, we show that for any round $r \geq r(\delta) + 1$, arm a^* defeats all other arms, i.e., $|D_r(a^*)| = |\mathcal{A}| - 1$. This would imply that $i_r = a^*$ and a^* is the champion. Consider any arm $j \in \mathcal{A} \setminus a^*$. Since a^* is compared to all active arms in round $r - 1 \geq r(\delta)$, we have

$$N_{a^*,j}(r-1) \geq q^{r-1} > \frac{8}{\Delta_{\min}^2} \cdot \log\left(2K^2 q_{r-1}\right),$$

where the second inequality is by Lemma 3.6 with $r - 1 \geq r(\delta)$. Now, by definition, we have

$$c_{a^*,j}(r-1) = \sqrt{\frac{2\log\left(2K^2 q_{r-1}\right)}{N_{a^*,j}(r-1)}} < \sqrt{\frac{2\log\left(2K^2 q_{r-1}\right)}{\frac{8}{\Delta_{\min}^2}\log\left(2K^2 q_{r-1}\right)}} = \frac{\Delta_{\min}}{2}.$$

Given this, it is easy to show that a^* defeats arm j in round r. Conditioned on $E(\delta)$, we know that $|\widehat{p}_{a^*,j}(r-1) - p_{a^*,j}| \leq c_{a^*,j}(r-1) \leq \frac{\Delta_{\min}}{2}$. Then, we have

$$\widehat{p}_{a^*,j}(r-1) \geq p_{a^*,j} - \frac{\Delta_{\min}}{2} = \frac{1}{2} + \Delta_j - \frac{\Delta_{\min}}{2} \geq \frac{1}{2} + \frac{\Delta_{\min}}{2} > \frac{1}{2} + c_{a^*,j}(r-1).$$

Therefore, $j \in D_r(a^*)$. It now follows that for any round $r \geq r(\delta) + 1$, arm a^* is the champion. □

Proof of Theorem 1.1. First, recall that in round r of `C2B`, any pair is compared $q_r = \lfloor q^r \rfloor$ times where $q = T^{1/B}$. Since $q^B = T$, `C2B` uses at most B rounds.

For the rest of proof, we fix $\delta > 0$. We now analyze the regret incurred by `C2B`, conditioned on events G and $E(\delta)$. Recall that $\mathbf{P}(G) \geq 1 - 1/T$ (Lemma 3.2), and $\mathbf{P}(E(\delta)) \geq 1 - \delta$ (Lemma 3.4). Thus, $\mathbf{P}(G \cap E(\delta)) \geq 1 - \delta - 1/T$. Let R_1 and R_2 denote the regret incurred before and after round $r(\delta)$ (see Definition 3.4) respectively.

Bounding R_1. This is the regret incurred up to (and including) round $r(\delta)$. We upper bound the regret by considering all pairwise comparisons every round $r \le r(\delta)$.

$$\begin{aligned} R_1 \;&\le\; K^2 \cdot \sum_{r \le r(\delta)} q_r \;\le\; K^2 \cdot \sum_{r \le r(\delta)} q^r \;\le\; 2K^2 \cdot q^{r(\delta)} \\ &\le 2K^2 \cdot \max\left\{ q \cdot 2A \log A \,,\, q^{C(\delta)+1} \right\}, \end{aligned}$$

where the last inequality uses Definition 3.4; recall $A = \frac{16}{\Delta_{\min}^2} \cdot \log(2K^2)$. Plugging in the value of $C(\delta) \le 1 + \frac{1}{2}\log_q(1/\delta)$, we obtain

$$R_1 \le O(K^2) \cdot \max\left\{ q \cdot \frac{\log K}{\Delta_{\min}^2} \cdot \log\left(\frac{\log K}{\Delta_{\min}}\right) \,,\, q^2 \sqrt{\frac{1}{\delta}} \right\}. \tag{5}$$

Bounding R_2. This is the regret in rounds $r \ge r(\delta) + 1$. By Lemma 3.7, arm a^* is the champion in all these rounds. So, the only comparisons in these rounds are of the form (a^*, j) for $j \in \mathcal{A}$.

Consider any arm $j \ne a^*$. Let T_j be the total number of comparisons that j participates in after round $r(\delta)$. Let r be the penultimate round that j is played in. We can assume that $r \ge r(\delta)$ (otherwise arm j will never participate in rounds after $r(\delta)$, i.e., $T_j = 0$). As arm j is *not* eliminated after round r,

$$\widehat{p}_{a^*,j}(r) \le \frac{1}{2} + \gamma_{a^*,j}(r).$$

Moreover, by $E(\delta)$, we have $\widehat{p}_{a^*,j}(r) \ge p_{a^*,j} - c_{a^*,j}(r)$ because $r \ge r(\delta) \ge C(\delta)$. So,

$$\frac{1}{2} + \Delta_j = p_{a^*,j} \le \widehat{p}_{a^*,j}(r) + c_{a^*,j}(r) \le \frac{1}{2} + \gamma_{a^*,j}(r) + c_{a^*,j}(r).$$

It follows that

$$\Delta_j \le \gamma_{a^*,j}(r) + c_{a^*,j}(r) \le \frac{3}{\sqrt{2}} \sqrt{\frac{\log(2K^2BT)}{N_{a^*,j}(r)}}$$

where the final inequality follows by definition of c and γ. On re-arranging, we get $N_{a^*,j}(r) \le \frac{9\log(2K^2BT)}{2\Delta_j^2}$. As $r+1$ is the last round that j is played in, and j is only compared to a^* in each round after $r(\delta)$,

$$T_j \;\le\; N_{a^*,j}(r+1) \;\le\; N_{a^*,j}(r) + 2q \cdot N_{a^*,j}(r) \;\le\; \frac{15q \cdot \log(2K^2BT)}{\Delta_j^2}.$$

The second inequality follows since j is compared to a^* in rounds r and $r+1$, and the number of comparisons in round $r+1$ is $\lfloor q^{r+1} \rfloor \le q \cdot (2q_r) \le 2q \cdot N_{a^*,j}(r)$. Adding over all arms j, the total regret accumulated beyond round $r(\delta)$ is

$$R_2 = \sum_{j \ne a^*} T_j \Delta_j \le \sum_{j \ne a^*} O\left(\frac{q \cdot \log(KT)}{\Delta_j}\right). \tag{6}$$

Combining (5) and (6), and using $q = T^{1/B}$, we obtain

$$\begin{aligned} R(T) \le O\left(T^{1/B} \cdot \frac{K^2 \log(K)}{\Delta_{\min}^2} \cdot \log\left(\frac{\log K}{\Delta_{\min}}\right)\right) \;&+\; O\left(T^{2/B} \cdot K^2 \cdot \sqrt{\frac{1}{\delta}}\right) \\ &+ \sum_{j \ne a^*} O\left(\frac{T^{1/B} \cdot \log(KT)}{\Delta_j}\right). \end{aligned}$$

This completes the proof Theorem 1.1. □

D Expected Regret Bound

In this section, we present the proof of Theorem 1.2. We first state the definitions needed in the proof. Let $F_X(\cdot)$ denote the cumulative density function (CDF) of a random variable X; that is, $F_X(x) = \mathbf{P}(X \leq x)$. The inverse CDF of X, denoted F_X^{-1}, is defined as $F_X^{-1}(z) = \inf\{x : \mathbf{P}(X \leq x) \geq z\}$ where $z \in [0, 1]$. We will use the identity $\mathbb{E}[X] = \int_0^1 F_X^{-1}(z)dz$.

Proof of Theorem 1.2. First, note that in round r of C2B, any pair is compared $q_r = \lfloor q^r \rfloor$ times where $q = T^{1/B}$. Since $q^B = T$, C2B uses at most B rounds.

Let $R(T)$ be the random variable denoting the regret incurred by C2B. By Theorem 1.1, we know that with probability at least $1 - \delta - 1/T$,

$$R(T) \leq O\left(T^{1/B} \cdot \frac{K^2 \log(K)}{\Delta_{\min}^2} \cdot \log\left(\frac{\log K}{\Delta_{\min}}\right)\right) + O\left(T^{2/B} \cdot K^2 \cdot \sqrt{\frac{1}{\delta}}\right) + \sum_{j \neq a^*} O\left(\frac{T^{1/B} \cdot \log(KT)}{\Delta_j}\right).$$

Thus, $F_{R(T)}^{-1}(1 - \delta - 1/T) \leq G(\delta)$ where

$$G(\delta) := A + O\left(T^{2/B} \cdot K^2 \cdot \sqrt{\frac{1}{\delta}}\right) + B$$

where to simplify notation we set $A = O\left(T^{1/B} \cdot \frac{K^2 \log(K)}{\Delta_{\min}^2} \cdot \log\left(\frac{\log K}{\Delta_{\min}}\right)\right)$ and $B = \sum_{j \neq a^*} O\left(\frac{T^{1/B} \cdot \log(KT)}{\Delta_j}\right)$. Using the identity for expectation of a random variable, we get

$$\begin{aligned}
\mathbb{E}[R(T)] &= \int_0^1 F_{R(T)}^{-1}(z)dz \\
&= \int_0^{1-\frac{1}{T}} F_{R(T)}^{-1}(z)dz + \underbrace{\int_{1-\frac{1}{T}}^{T} F_{R(T)}^{-1}(z)dz}_{\leq T \cdot \frac{1}{T} = 1} \\
&\leq \int_0^{1-\frac{1}{T}} F_{R(T)}^{-1}(z)dz \ + \ 1 \\
&= \int_{1-\frac{1}{T}}^{0} F_{R(T)}^{-1}\left(1 - \delta - \frac{1}{T}\right)(-d\delta) \ + \ 1 \\
&\leq \int_0^{1-\frac{1}{T}} G(\delta)d\delta \ + \ 1 \\
&\leq A + O\left(T^{2/B} \cdot K^2\right) + B + 1
\end{aligned}$$

where the fourth equality follows by setting $1 - q - 1/T = \delta$ and the final inequality follows since $\int_0^1 \left(\frac{1}{\delta}\right)^{1/2} \leq 2$. Thus,

$$\mathbb{E}[R(T)] \leq O\left(T^{1/B} \cdot \frac{K^2 \log(K)}{\Delta_{\min}^2} \cdot \log\left(\frac{\log K}{\Delta_{\min}}\right)\right) + O\left(T^{2/B} \cdot K^2\right) + \sum_{j \neq a^*} O\left(\frac{T^{1/B} \cdot \log(KT)}{\Delta_j}\right).$$

This completes the proof of Theorem 1.2. □

E The Batched Algorithm with KL-based Elimination Criterion

In this section, we modify C2B to use a Kullback-Leibler divergence based elimination criterion. We provide a complete description of the algorithm, denoted C2B-KL, in Algorithm 2. In what follows, we highlight the main differences of C2B-KL from C2B. Recall the following notation. We use $\mathcal{A}$ to denote the current set of *active* arms; i.e., the arms that have not been eliminated. We use index r for rounds or batches. If pair (i, j) is compared in round r, it is compared $q_r = \lfloor q^r \rfloor$ times where $q = T^{1/B}$. We define the following quantities at the *end* of each round r:

- $N_{i,j}(r)$ is the total number of times the pair (i, j) has been compared.
- $\widehat{p}_{i,j}(r)$ is the frequentist estimate of $p_{i,j}$, i.e.,

$$\widehat{p}_{i,j}(r) = \frac{\#\, i \text{ wins against } j \text{ until end of round } r}{N_{i,j}(r)}\,. \tag{7}$$

- A confidence-interval radius for each (i, j) pair:

$$c_{i,j}(r) = \sqrt{\frac{2\log(2K^2 q_r)}{N_{i,j}(r)}}$$

- We define a term $I_j(r)$ which, at a high-level, measures how unlikely it is for j to be the Condorcet winner at the end of batch r:

$$I_j(r) = \sum_{i:\widehat{p}_{i,j}(r)\geq\frac{1}{2}} D_{\text{KL}}\left(\widehat{p}_{i,j}(r), \frac{1}{2}\right) \cdot N_{i,j}(r),$$

  where $D_{\text{KL}}(p, q)$ denotes the Kullback–Leibler divergence between two Bernoulli distributions: $B(p)$ and $B(q)$. We define $I^*(r) = \min_{j\in\mathcal{A}} I_j(r)$.

The B-round algorithm, C2B-KL, proceeds exactly as C2B. The only change is in the *elimination criterion*, which we describe next.

Elimination Criterion. In round r, if, for any arm j, we have $I_j(r) - I^*(r) > \log(T) + f(K)$, then j is eliminated from $\mathcal{A}$. Here $f(K)$ is a non-negative function of K, independent of r.

The main result of this section is to show that C2B-KL achieves the following guarantee.

Theorem E.1. *For any integer $B \geq 1$, there is an algorithm for the K-armed dueling bandit problem that uses at most B rounds. Furthermore, for any $\delta > 0$, with probability at least $1 - \delta - \frac{1}{T} \cdot e^{K\log(C)-f(K)}$, where C is some constant (see Lemma E.2), its regret under the Condorcet condition is at most*

$$R(T) \leq O\left(T^{1/B} \cdot \frac{K^2\log(K)}{\Delta_{\min}^2} \cdot \log\left(\frac{\log K}{\Delta_{\min}}\right)\right) + O\left(T^{2/B} \cdot K^2 \cdot \sqrt{\frac{1}{\delta}}\right) + \sum_{j\neq a^*} O\left(\frac{T^{1/B}\cdot\log(T)}{\Delta_j}\right) + \sum_{j\neq a^*} O\left(\frac{T^{1/B}\cdot f(K)}{\Delta_j}\right)$$

Remark. Setting $f(K) > K\log(C)$, we get the same asymptotic expected regret bound as in Theorem 1.2. Following [35], we set $f(K) = 0.3K^{1.01}$ in our experiments.

We require the following result in the proof of Theorem E.1.

Fact E.1. *For any μ and μ_2 satisfying $0 < \mu_2 < \mu < 1$. Let $C_1(\mu, \mu_2) = (\mu - \mu_2)^2/(2\mu(1 - \mu_2))$. Then, for any $\mu_3 \leq \mu_2$,*

$$D_{KL}(\mu_3, \mu) - D_{KL}(\mu_3, \mu_2) \geq C_1(\mu, \mu_2) > 0.$$

The high-level outline of the analysis is exactly the same as that of C2B. For completeness, we provide the analysis in the following section; however, we skip the proofs of lemmas that follow from the analysis of C2B.

Algorithm 2 C2B-KL

Input: Arms $\mathcal{B}$, time-horizon T, integer $B \geq 1$
active arms $\mathcal{A} \leftarrow \mathcal{B}$, $r \leftarrow 1$, emprical probabilities $\widehat{p}_{i,j}(0) = \frac{1}{2}$ for all $i, j \in \mathcal{B}^2$
while number of comparisons $\leq T$ **do**
  if $\mathcal{A} = \{i\}$ for some i **then** play (i, i) for remaining trials
  $D_r(i) \leftarrow \{j \in \mathcal{A} : \widehat{p}_{i,j}(r-1) > \frac{1}{2} + c_{i,j}(r-1)\}$
  $i_r \leftarrow \arg\max_{i \in \mathcal{A}} |D_r(i)|$
  for $i \in \mathcal{A} \setminus \{i_r\}$ **do**
    if $i \in D_r(i_r)$ **then**
      compare (i_r, i) for q_r times
    else
      for each $j \in \mathcal{A}$, compare (i, j) for q_r times
  compute $\widehat{p}_{i,j}(r)$ values
  if $\exists j : I_j(r) - I^*(r) > \log(T) + f(K)$ **then**
    $\mathcal{A} \leftarrow \mathcal{A} \setminus \{j\}$
    $r \leftarrow r + 1$

E.1 The Analysis

In this section, we prove the high-probability regret bound for C2B-KL. Recall that $q = T^{1/B}$, and that $q \geq 2$. We first show that, with high probability, a^* is not eliminated during the execution of the algorithm. The following lemma formalizes this.

Lemma E.2. *Let G denote the event that the best arm a^* is not eliminated during the execution of* C2B-KL*. We can bound the probability of $\overline{G}$ as follows.*

$$\mathbf{P}(\overline{G}) \leq \frac{1}{T} \cdot e^{K \log(C) - f(K)},$$

where $C = \max_j C(j) + 1$, is a constant, with $C(j) = \left(\frac{1}{e^{D_{KL}\left(p_{j,a^*},1/2\right)} - 1} + \frac{e^{C_1\left(p_{a^*,j},1/2\right)}}{\left(e^{C_1\left(p_{a^*,j},1/2\right)} - 1\right)^2} \right)$.

Proof. Let n_j denote the number of times a^* and j are compared. Let $\widehat{p}_{a^*,j}(n_j)$ denote the frequentist estimate of $p_{a^*,j}$ when a^* and j are compared n_j times (we will abuse notation and use $\widehat{p}_{a^*,j}$ when n_j is clear from context). Let $S \in 2^{[K] \setminus \{a^*\}} \setminus \emptyset$, and consider vector $\{n_j \in \mathbb{N} : j \in S\}$. We define $A = \sum_{j \in S} D_{\text{KL}}\left(\widehat{p}_{j,a^*}, 1/2\right) \cdot n_j$. Let $D(S; \{n_j : j \in S\})$ denote the event that a^* and j are compared n_j times and $\widehat{p}_{a^*,j} \leq 1/2$ for all $j \in S$, and that $A > \log(T) + f(K)$. The probability of this event upper bounds the probability that a^* is eliminated (as per our elimination criterion) when a^* and j are compared n_j times, and $\widehat{p}_{a^*,j} \leq 1/2$ for all $j \in S$. We will show that

$$\mathbf{P}(D(S; \{n_j : j \in S\})) \leq \frac{e^{-f(K)}}{T} \prod_{j \in S} \left(e^{-n_j D_{\text{KL}}\left(p_{j,a^*},1/2\right)} + n_j e^{C_1\left(p_{j,a^*},1/2\right)} \right) \tag{8}$$

where $C_1(\mu_1, \mu_2) = (\mu_1 - \mu_2)^2/(2\mu_1(1-\mu_2))$. Using the above, we first show that by taking a union bound over all $S \in 2^{[K]\setminus\{a^*\}} \setminus \emptyset$ and $\{n_j : j \in S\}$, we obtain the final result. We have

$$
\begin{aligned}
\mathbf{P}(\overline{G}) &\leq \sum_{S \in 2^{[K]\setminus\{a^*\}}\setminus\emptyset} \sum_{n_j \in \mathbb{N}^{|S|}} \mathbf{P}(D(S; \{n_j : j \in S\})) \\
&\leq \sum_{S \in 2^{[K]\setminus\{a^*\}}\setminus\emptyset} \sum_{n_j \in \mathbb{N}^{|S|}} \frac{e^{-f(K)}}{T} \prod_{j \in S} \left(e^{-n_j D_{\mathrm{KL}}\left(p_{j,a^*}, 1/2\right)} + n_j e^{C_1\left(p_{j,a^*}, 1/2\right)} \right) \\
&= \frac{e^{-f(K)}}{T} \sum_{S \in 2^{[K]\setminus\{a^*\}}\setminus\emptyset} \prod_{j \in S} \sum_{n_j \in \mathbb{N}} \left(e^{-n_j D_{\mathrm{KL}}\left(p_{j,a^*}, 1/2\right)} + n_j e^{C_1\left(p_{j,a^*}, 1/2\right)} \right) && (9) \\
&= \frac{e^{-f(K)}}{T} \sum_{S \in 2^{[K]\setminus\{a^*\}}\setminus\emptyset} \prod_{j \in S} \left(\frac{1}{e^{D_{\mathrm{KL}}\left(p_{j,a^*}, 1/2\right)} - 1} + \frac{e^{C_1\left(p_{j,a^*}, 1/2\right)}}{\left(e^{C_1\left(p_{j,a^*}, 1/2\right)} - 1\right)^2} \right) && (10) \\
&\leq \frac{e^{-f(K)}}{T} \sum_{S \in 2^{[K]\setminus\{a^*\}}\setminus\emptyset} (C-1)^{|S|} \leq \frac{e^{-f(K)}}{T} \cdot C^K && (11) \\
&= \frac{1}{T} \cdot e^{K \log(C) - f(K)}
\end{aligned}
$$

where (9) follows by swapping the order of summation and multiplication, (10) uses $\sum_{n=1}^{\infty} e^{-nx} = 1/(e^x - 1)$ and $\sum_{n=1}^{\infty} n e^{-nx} = e^x/(e^x-1)^2$, and (11) follows by letting $C(j) = \left(\frac{1}{e^{D_{\mathrm{KL}}\left(p_{j,a^*}, 1/2\right)} - 1} + \frac{e^{C_1\left(p_{j,a^*}, 1/2\right)}}{\left(e^{C_1\left(p_{j,a^*}, 1/2\right)} - 1\right)^2} \right)$, $C = \max_j C(j) + 1$ and the binomial theorem. To complete the proof, we need to prove (8).

For the remainder of this proof, we fix $S \in 2^{[K]\setminus\{a^*\}} \setminus \emptyset$, and vector $\{n_j \in \mathbb{N} : j \in S\}$. Observe that

$$
\mathbf{P}(D(S; \{n_j : j \in S\})) = \mathbf{P}\left(A > \log(T) + f(K)\right) = \mathbf{P}\left(T < e^{-f(K)} \cdot e^A\right)
$$

where we defined $A = \sum_{j \in S} D_{\mathrm{KL}}\left(\widehat{p}_{j,a^*}, 1/2\right) \cdot n_j$. By Markov's inequality, we have

$$
\mathbf{P}\left(e^{-f(K)} \cdot e^A > T\right) \leq \frac{\mathbb{E}[e^{-f(K)} \cdot e^A]}{T} = \frac{e^{-f(K)}}{T} \cdot \mathbb{E}[e^A] \tag{12}
$$

where the last equality follows since $f(K)$ is constant (with respect to $\{n_j\}$ values). So, it suffices to bound $\mathbb{E}[e^A]$. Towards this end, we define the following term:

$$
P_j(x_j) = \mathbf{P}\left(\widehat{p}_{j,a^*} \geq \frac{1}{2} \text{ and } D_{\mathrm{KL}}\left(\widehat{p}_{j,a^*}, \frac{1}{2}\right) \geq x_j\right).
$$

Then, we have

$$
\begin{aligned}
\mathbb{E}[e^A] &= \int_{\{x_j\}\in[0,\log(2)]^{|S|}} \exp\left(\sum_{j\in S} n_j x_j\right) \prod_{j\in S} d(-P_j(x_j)) \\
&= \prod_{j\in S} \int_{x_j\in[0,\log 2]} e^{n_j x_j} d(-P_j(x_j)) && (13)\\
&= \prod_{j\in S} \left([-e^{n_j x_j} P_j(x_j)]_0^{\log(2)} + \int_{x_j\in[0,\log(2)]} n_j e^{n_j x_j} P_j(x_j) dx_j \right) && (14)\\
&= \prod_{j\in S} \left(P_j(0) + \int_{x_j\in[0,\log(2)]} n_j e^{n_j x_j} P_j(x_j) dx_j \right) \\
&\le \prod_{j\in S} \left(e^{-n_j D_{\mathrm{KL}}\left(p_{j,a^*},1/2\right)} + \int_{x_j\in[0,\log(2)]} n_j e^{n_j x_j} e^{-n_j\left(x_j + C_1\left(p_{j,a^*},1/2\right)\right)} dx_j \right) && (15)\\
&= \prod_{j\in S} \left(e^{-n_j D_{\mathrm{KL}}\left(p_{j,a^*},1/2\right)} + \int_{x_j\in[0,\log(2)]} n_j e^{C_1\left(p_{j,a^*},1/2\right)} dx_j \right) \\
&\le \prod_{j\in S} \left(e^{-n_j D_{\mathrm{KL}}\left(p_{j,a^*},1/2\right)} + n_j e^{C_1\left(p_{j,a^*},1/2\right)} \right)
\end{aligned}
$$

where (13) follows from the independence of the comparisons. We obtain (14) by applying integration by parts, (15) follows from the Chernoff bound and Fact E.1; here $C_1(\mu_1,\mu_2) = (\mu_1-\mu_2)^2/(2\mu_1(1-\mu_2))$, and the final inequality follows by observing that $\int_{x_j\in[0,\log(2)]} n_j e^{C_1\left(p_{j,a^*},1/2\right)} dx_j = n_j e^{C_1\left(p_{j,a^*},1/2\right)} \cdot \int_{x_j\in[0,\log(2)]} dx_j = n_j e^{C_1\left(p_{j,a^*},1/2\right)} \log(2)$. Note that log refers to the natural logarithm, so we have $\log(2) \le 1$. Combined with (12), this completes the proof of (8). □

E.1.1 High-probability Regret Bound

We now prove Theorem E.1. Fix any $\delta > 0$. We first define event $E(\delta)$ as before.

Definition E.1 (Event $E(\delta)$)**.** *An estimate $\widehat{p}_{i,j}(r)$ in batch r is **weakly-correct** if $|\widehat{p}_{i,j}(r) - p_{i,j}| \le c_{i,j}(r)$. Let $C(\delta) := \lceil \frac{1}{2}\log_q(1/\delta)\rceil$. We say that event $E(\delta)$ occurs if for each batch $r \ge C(\delta)$, every estimate is weakly-correct.*

The next lemma shows that $E(\delta)$ occurs with probability at least $1-\delta$. Since $E(\delta)$ does not depend on the elimination criterion, its proof follows from the analysis of `C2B`.

Lemma E.3. *For all $\delta > 0$, we have*

$$\mathbf{P}(\neg E(\delta)) \;=\; \mathbf{P}\left(\exists r \ge C(\delta), i, j : |\widehat{p}_{i,j}(r) - p_{i,j}| > c_{i,j}(r)\right) \;\le\; \delta.$$

As before, we analyze our algorithm under both events G and $E(\delta)$. Recall that, under event G, the best arm a^* is not eliminated. *Conditioned on these*, we next show:

- The best arm, a^*, is *not defeated* by any arm i in any round $r > C(\delta)$ (Lemma E.4).
- Furthermore, there exists a round $r(\delta) \ge C(\delta)$ such that arm a^* defeats *every other arm*, in every round after $r(\delta)$ (Lemma E.6).

We re-state the formal lemmas next.

Lemma E.4. *Conditioned on G and $E(\delta)$, for any round $r > C(\delta)$, arm a^* is not defeated by any other arm, i.e., $a^* \notin \cup_{i\neq a^*} D_r(i)$.*

To proceed, we need the following definitions.

Definition E.2. *The candidate i_r of round r is called the **champion** if $|D_r(i_r)| = |\mathcal{A}|-1$; that is, if i_r defeats every other active arm.*

Definition E.3. *Let $r(\delta) \geq C(\delta) + 1$ be the smallest integer such that*

$$q^{r(\delta)} \geq 2A \log A, \qquad \textit{where } A := \frac{32}{\Delta_{\min}^2} \cdot \log(2K^2).$$

We use the following inequality based on this choice of $r(\delta)$.

Lemma E.5. *The above choice of $r(\delta)$ satisfies*

$$q^r > \frac{8}{\Delta_{\min}^2} \cdot \log\left(2K^2 q_r\right), \qquad \forall r \geq r(\delta).$$

Then, we have the following.

Lemma E.6. *Conditioned on G and $E(\delta)$, the best arm a^* is the champion in every round $r > r(\delta)$.*

We are now ready to prove Theorem E.1.

Proof of Theorem E.1. First, recall that in round r of C2B, any pair is compared $q_r = \lfloor q^r \rfloor$ times where $q = T^{1/B}$. Since $q^B = T$, C2B uses at most B rounds.

For the rest of proof, we fix $\delta > 0$. We now analyze the regret incurred by C2B, conditioned on events G and $E(\delta)$. Recall that $\mathbf{P}(G) \geq 1 - \frac{1}{T} \cdot e^{K \log(C) - f(K)}$ (Lemma E.2), and $\mathbf{P}(E(\delta)) \geq 1 - \delta$ (Lemma E.3). Thus, $\mathbf{P}(G \cap E(\delta)) \geq 1 - \delta - \frac{1}{T} \cdot e^{K \log(C) - f(K)}$. Let R_1 and R_2 denote the regret incurred before and after round $r(\delta)$ (see Definition E.3) respectively.

Bounding R_1. We can bound R_1 as in the proof of Theorem 1.1; so, we get

$$R_1 \leq O(K^2) \cdot \max\left\{ q \cdot \frac{\log K}{\Delta_{\min}^2} \cdot \log\left(\frac{\log K}{\Delta_{\min}}\right),\ q^2 \sqrt{\frac{1}{\delta}} \right\}. \tag{16}$$

Bounding R_2. This is the regret in rounds $r \geq r(\delta) + 1$. By Lemma E.6, arm a^* is the champion in all these rounds. So, the only comparisons in these rounds are of the form (a^*, j) for $j \in \mathcal{A}$.

Consider any arm $j \neq a^*$. Let T_j be the total number of comparisons that j participates in after round $r(\delta)$. Let r be the penultimate round that j is played in. We can assume that $r \geq r(\delta)$ (otherwise arm j will never participate in rounds after $r(\delta)$, i.e., $T_j = 0$). As arm j is *not* eliminated after round r,

$$I_j(r) - I^*(r) \leq \log(T) + f(K).$$

By Lemma E.6, $I^*(r) = 0$ (since a^* is the *champion*, the summation is empty). So, we have $I_j(r) \leq \log(T) + f(K)$. Observe that

$$I_j(r) \geq D_{\text{KL}}\left(\widehat{p}_{a^*,j}(r), \frac{1}{2}\right) N_{a^*,j}(r) \tag{17}$$

We can lower bound $D_{\text{KL}}\left(\widehat{p}_{a^*,j}(r), \frac{1}{2}\right)$ as follows.

$$D_{\text{KL}}\left(\widehat{p}_{a^*,j}(r), \frac{1}{2}\right) \geq \left(\widehat{p}_{a^*,j}(r) - \frac{1}{2}\right)^2 \geq \left(p_{a^*,j} - c_{a^*,j}(r) - \frac{1}{2}\right)^2 \geq \left(\frac{\Delta_j}{2}\right)^2$$

where the first inequality follows from Pinsker's inequality, the second inequality uses Lemma E.3 and the final inequality uses the fact that $c_{a^*,j}(r) \leq \frac{\Delta_{\min}}{2}$, which follows by the choice of $r(\delta)$. Plugging this into (17), we get

$$\frac{\Delta_j^2}{4} \cdot N_{a^*,j}(r) \leq \log(T) + f(K)$$

which on re-arranging gives

$$N_{a^*,j}(r) \leq \frac{4(\log(T) + f(K))}{\Delta_j^2}.$$

As $r+1$ is the last round that j is played in, and j is only compared to a^* in each round after $r(\delta)$,

$$T_j \;\le\; N_{a^*,j}(r+1) \;\le\; N_{a^*,j}(r) + 2q \cdot N_{a^*,j}(r) \;\le\; \frac{12q \cdot (\log(T) + f(K))}{\Delta_j^2}.$$

The second inequality follows since j is compared to a^* in rounds r and $r+1$, and the number of comparisons in round $r+1$ is $\lfloor q^{r+1} \rfloor \le q \cdot (2q_r) \le 2q \cdot N_{a^*,j}(r)$. Adding over all arms j, the total regret accumulated beyond round $r(\delta)$ is

$$R_2 = \sum_{j \neq a^*} T_j \Delta_j \le \sum_{j \neq a^*} O\left(\frac{q \cdot (\log(T) + f(K))}{\Delta_j}\right). \tag{18}$$

Combining (16) and (18), and using $q = T^{1/B}$, we obtain

$$R(T) \le O\left(T^{1/B} \cdot \frac{K^2 \log(K)}{\Delta_{\min}^2} \cdot \log\left(\frac{\log K}{\Delta_{\min}}\right)\right) \quad + O\left(T^{2/B} \cdot K^2 \cdot \sqrt{\frac{1}{\delta}}\right) + \sum_{j \neq a^*} O\left(\frac{T^{1/B} \cdot \log(T)}{\Delta_j}\right) + \sum_{j \neq a^*} O\left(\frac{T^{1/B} \cdot f(K)}{\Delta_j}\right)$$

This completes the proof Theorem E.1. □

F Hardware Specification for Computational Experiments

We conducted our computations using C++ and Python 2.7 with a 2.3 Ghz Intel Core $i5$ processor and 16 GB 2133 MHz LPDDR3 memory.