Reviewer #1

Re. statement of lower bound (Theorem 1): The theorem statement should say "there exists a GCC choice model with $\Delta^{\mathrm{GCC}}_{\min} = \Delta$ such that...". We will correct this in the final version if accepted.

Re. structured bandits lower bound: Thanks for pointing us to this reference (which we will certainly include). Although our choice bandits problem can be cast as a structured bandit problem, this existing lower bound is in terms of the solution of an LP, and it is not clear how this solution relates to the gaps in the underlying choice model instance. The main novelty in our lower bound is to construct a distribution over hard instances which allows us to identify a fundamental gap parameter quantifying the regret any algorithm must incur.

Re. choice of parameter C in our algorithm (and Theorem 2): The parameter C can indeed be chosen in a problem-independent manner. In particular, setting $C = T^4$ suffices for Theorem 2 to hold, giving a regret upper bound of $O(\log(TC)) = O(\log(T^5)) = O(\log(T))$. (If T is not known, we can use the doubling trick.) To see why the choice $C = T^4$ suffices, note that in the proof of Theorem 2, we mention it is sufficient for C to be larger than a term C' (defined at the end of page 11 in the supplementary material). While the term C' is problem-dependent, we do not actually need to know its exact value to set C larger than it. In particular, note that C' is upper bounded by $\left(1/\Delta^{\mathrm{GCC}}_{\min}\right)^4$; moreover, in order to obtain any non-trivial upper bound for our algorithm, $\Delta^{\mathrm{GCC}}_{\min}$ has to be larger than $1/T$. Hence, either C' is upper bounded by T^4, or the instance is too hard to allow any non-trivial upper bound. Therefore, setting $C = T^4$ suffices to ensure $C \geq C'$ whenever the instance is not already too hard. We actually believe setting $C = T^4$ may be somewhat pessimistic (it arises from taking a union bound over all possible states of the algorithm in Lemma 2) – indeed, in our experiments, we set $C = 1$ for all datasets, and our algorithm still demonstrates sublinear regret with this choice – but it certainly suffices, and the regret bound with $C = T^4$ is at most a constant factor 5 times what one might get with $C = 1$ if the regret bound holds in that case. We will add a discussion on this in the final version if accepted.

Reviewer #2

Thanks for the suggestions for making the paper more self-contained, and also for the reference on dynamic learning in the MNL model. We will incorporate all these suggestions and include this reference in the final version if accepted.

Re. high-level goal (and notion of regret): The goal is to identify the 'best' arm ***while also playing good/competitive assortments during the exploration phase***. This is similar to dueling bandit settings where the goal is to identify the best arm while also playing good/competitive pairs during the exploration phase; we are *not* working in the pure exploration setting, where all assortments/pairs in the exploration phase incur uniform cost. Accordingly, our notion of regret penalizes an algorithm for playing assortments that are not 'competitive'. (E.g. in the case of the MNL model, if $v_i \approx 0$ for some arm i, then an algorithm incurs a lot of regret for playing i.) We will certainly clarify this.

Re. real-world applications: Essentially, we are interested in applications where the goal is to identify the 'best' item/product (or more generally, in future work, a collection of 'good' items/products), ***while also serving well the users with whom the learning system interacts during the exploration phase in order to identify such items***. Applications could include online advertising, recommender systems, and online ranker evaluation for information retrieval. The latter can be viewed as a generalization of the ranker evaluation application considered by Yue and Joachims (ICML 2009), in which the goal was to identify the better of two ranking systems by interleaving their results and using a dueling bandit formulation; our setting would allow extending this to the identification of the best among $k \geq 2$ ranking systems by 'multi-leaving' their results, all while still presenting acceptable/good results to the users who are using the system during the exploration phase. We will be happy to add some comments on this as well.

Reviewer #3

Re. lower bound (Theorem 1): Yes, your intuition is correct. Please also see our response to Reviewer #1 above.

Re. upper bound (Theorem 2): You are right; the $O(n^2 \log n)$ term suppresses some problem dependent terms. The only reason for this was to simplify the exposition and to emphasize our algorithm is asymptotically order optimal as this term does not increase with T. The precise value of this term is given by $O(n \log n \times \sum_{i \neq i^*} \frac{1}{d(1/2, P^{\mathrm{GCC}}_{i^* i})})$, where $d(\cdot, \cdot)$ is the KL-divergence. We have seen a similar approach to writing regret upper bounds in other papers (e.g. RMED paper), but we agree it would be good to be more precise upfront; we will certainly add a note on this.

Re. parameter C: This can be chosen in a problem-independent manner. Please see our response to Reviewer #1.

Re. discussion after Theorem 2: Our intention was to highlight that the bound is not *directly* a function of k, but rather has a more subtle dependence through the problem parameters. When k increases, we consider larger sets and a broader problem instance, and hence, the terms $\Delta^{\mathrm{GCC}}_{\max}$ and $\Delta^{\mathrm{GCC}}_{\min}$ change. There is indeed a gap in the upper and lower bounds in the general GCC case, but even in this case, the dependence on other parameters such as n and T is asymptotically order-optimal. We will clarify these points in the final version if accepted.