We would like to thank the reviewers for their thorough reading of the paper and their comments. Below we address some of the main points. We will address all comments in the final version.

Technical contribution and novelty: All reviewers seem to agree that the problem we address is important and timely, and that our results are interesting and useful. We believe that the fact that we achieve these results in a simple way should not be held against us. (Ideally, it should not be the case that a paper achieving the same results with a more complex solution would have a higher chance of being accepted.) On the contrary, we think that, as Reviewer 4 pointed out, the simplicity of the reductions that we have developed, which allow for the incorporation of fairness constraints easily into existing frameworks without significant changes, is one of the strengths of our work, and it makes our algorithms easy to apply in practice.

In terms of novelty of the work and theoretical/technical contribution, we would like to clarify several points.

• The fairness constraints do not themselves constitute a matroid and the natural reduction to submodular maximization subject to matroid constraints does not work. Our reduction relies on the novel notion of *extendable* sets and the observation that the family of extendable sets forms a matroid. Although this observation is not difficult to prove once stated, the idea itself is far from intuitively obvious.
• Theorem 4.3 follows directly from our reduction (Theorem 4.2), but this is not the case for Theorem 4.4 (the main result in the monotone case). Achieving the faster running time in Theorem 4.4 requires more work. In works on matroid-constrained submodular maximization, the complexity is given in terms of the number of matroid queries. In the case of the matroid of extendable sets, a naive implementation of such a query would create dependencies on the number of colors (the value C) for the running time. Therefore we designed an implementation using efficient data structures that handles such queries in $\widetilde{O}(1)$ time for each element of the stream (see Appendices B.2 and D).
• The *excess ratio* concept we introduce in this work is novel. Without it, one cannot apply the previous ideas to the non-monotone case. This concept may also be of independent interest for various classic problems in the streaming setting subject to fairness constraints, e.g., matching, vertex cover, prophet, or secretary problems.
• One of the main contributions of this work is the hardness result (Theorem 5.1), showing that our non-monotone algorithm is asymptotically tight. This also confirms the theoretical importance of the *excess ratio*. This result is novel, and its proof is fairly involved (see Appendix E.2).

Reviewer 1: We hope that the points above address your concerns regarding the theoretical contribution of our work.
• *"I would be interested in a bound between the fair and unfair instances."* From a theoretical point of view, one can come up with a simple example where the optimum "fair" solution has value zero, while the optimum "unfair" solution has value one. Hence, the cost of fairness is in general unbounded, but as you correctly mentioned it is indeed highly application dependent. To account for this, we ran experiments for 4 applications, with different datasets and parameters.

Reviewer 2: We hope that the points above address your concerns regarding the novelty of our work.
• *"Please clarify that in Figure 1[c,f,i] the y axis is the number of oracle calls at iteration k"* The y-axis corresponds to the total number of oracle calls for a given cardinality constraint k. We will emphasize this in the final version.
• *"How much do the sets returned by these algorithms differ?"* In some cases they differ a lot. As reported in Figures 1-3, the solutions returned by the "unfair" algorithms violate the fairness constraints significantly, and accordingly they must also differ significantly from the solutions returned by our proposed algorithms.
• *"Do the results assume the number of colors C is constant?"* No, we do not make such an assumption; the number of colors can be any value between 1 and n (length of the stream). Surprisingly, the running time and memory consumption are independent from the value of C. We achieve this by utilizing efficient data structures and update techniques. The details are provided in the appendix.

Reviewer 3: We hope that the points above address your concerns regarding the technical contribution of our work.
• *"How is it possible that Upper-bounds makes more calls? Isn't it just the FKK algorithm without the extendability requirement?"* This is a great point. Notice that in the case when we have only upper-bounds, there are more feasible elements that we need to check for swapping compared to the case where we have both upper and lower bounds. Therefore, Upper-bounds can require more oracle calls.

Reviewer 4: Thank you for the insightful comments and constructive feedback. We will adjust the tone of both the introduction and abstract, and correct the errors in the prior work section. Regarding the limitation illustrated with the college admission example: Note that Algorithm 2 will admit students, from the beginning, differently compared to an algorithm without fairness constraints, due to the extendability constraints. Moreover, we find that in practice, augmentation is not required, as the cardinality constraint is reached by the time the stream is finished.