Response to NeurIPS 2020 Reviews #2822 *Adversarially Robust Streaming Algorithms via Differential Privacy*

We thank all four reviewers for their time and comments. Their suggestions will help us clarify the contributions of our work as we incorporate them in the next revision of our paper.

We view our work as providing both technical and conceptual contributions. The technical contribution is that we are the first to present adversarially robust streaming algorithms that work in the general turnstile model (where both positive and negative updates are allowed). Specifically, the previous work of Ben-Eliezer et al had space complexity that grows linearly with λ (the flip number). In the turnstile model, λ can be as big as the length of the stream, and hence the algorithms of Ben-Eliezer et al do not provide meaningful (worst-case) bounds. In contrast, our space complexity only grows as $\sqrt{\lambda}$, and hence, our algorithm has sublinear space also in the general turnstile model.

The conceptual contribution of our work is that it formally connects the fields of adaptive data analysis and differential privacy with the field of adversarially robust streaming. In particular, to the best of our knowledge, we are the first to use differential privacy in order to protect the internal randomness of the algorithm.

In the following, we respond to several specific points raised by the reviewers.

Reviewer #1: *"One weakness is the lack of lower bounds"*

We agree that our work does not close the door on this question. Our work is the first to show sub-linear space in the turnstile model for adversarially robust streaming, and suggests interesting future directions – in both upper and lower bounds.

Reviewer #2: *"benefit from more intuition on why the differential privacy theorems apply"*
"In Theorem 2.7, why is there no dependence on $\log|X|$ *and on* $1/\varepsilon^3$ *as in Bassily et al?"*

We will improve the presentation w.r.t. introducing differential privacy. Theorem 2.7 appears in Bassily et al as Theorem 7.2 (see their arXiv version).

Reviewer #2: *"Why can the generalization bound be used in the proof of Lemma 3.2? Is it because of post-processing?"*

Yes, it's because of post-processing. We will make it clear in the next revision of our paper.

Reviewer #2 and #4: *"The paper focuses on improving 1/epsilon factors"*
"Quantitatively, the 1/epsilon improvements are incremental"
"For constant eps, it is not clear how good the improvement is"

Our technical focus is on improving the dependency in λ (the flip number) from linear to square root, which allows us to present the first adversarially robust algorithm for the general turnstile model (see paragraph at the beginning of this rebuttal). In the turnstile model we could have that $\lambda = \Theta(m)$, where m is the length of the stream, at which case previous works do not obtain meaningful bounds (even when the approximation parameter is constant). We use space at most roughly $\sqrt{\lambda}$, which is sublinear, and therefore obtain the first algorithm for the turnstile case. As a by-product, in some parameter regimes, we also improve over existing results in the insertion-only model (in terms of the approximation parameter), but this is not our focus.

Reviewer #2 and #4: *"Not entirely convinced the presentation and results are catered toward the ML audience"*
"NeurIPS may not be the right venue"

Our work applies differential privacy and generalization bounds to make streaming algorithms robust to adversarial attacks and feedback loops (in which the value reported by the algorithm affects future updates). Each of these topics, namely, differential privacy, generalization, adversarial attacks, and streaming and sketching, has been of interest to the ML community and addressed in previous NeurIPS conferences. In particular, our work has a large intersection with the field of adaptive data analysis, which is one of the areas that appeared in the call for papers (under Algorithms).

Our idea of using privacy as a tool to protect against adversarial attacks on the randomness of the algorithm may be applicable whenever a randomized ML model that reports continuously is exposed to a dangerous feedback loop or malicious users. We will make this connection more explicit in the next revision of our paper.