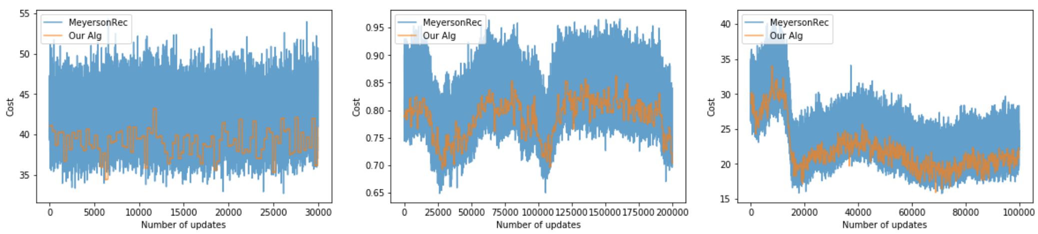
We thank the reviewers for the thorough reviews.

Reviewer #1 on the improvement over naive algorithms and Reviewer #2 on lower bounds: We would like to point out that for k-median and k-means in general metric spaces, there is an $\Omega(nk)$ lower bound for computing an $O(1)$-approximation that holds even for offline algorithms (see [2]). Thus, the naive algorithms achieve a total running time of $\Omega(n(nk))$ – $\Omega(n^2k)$ (by recomputing each time), whereas our algorithm achieves a total running time of $O(n(n + k^2)polylogn)$ $O((n^2 + nk^2)polylogn)$ which is better for any $k = \omega(polylogn)$. For practical cases where k is often n^c for some constant c, this is a significant improvement. Thus, our algorithm indeed has a better running time than the naive algorithms. Notice that these bounds are tight up to $O(polylogn)$ factors of $k = O(\sqrt{n})$, as inserting a new point in a general metric requires $O(n)$ time to describe the distances to other points. We will include this discussion in the next version of our paper.

Reviewer #1 and #3 on the experimental section: We acknowledge that the write-up of the experimental section is not optimal, we realized that the blue line was hidden by the orange line, the new figures for the cost vs the number of updates for our algorithms vs MeyersonRec for USC, Twitter and Covtype are the following (in the preceeding order)



We will also add a description of the k-median/means algorithm within the first 8 pages.

Reviewer #2 on problem justification: The study of online clustering dates back to the early 2000s. In many practical scenarios, datasets that we would like to cluster are dynamic, for example webpages, search queries, news articles, social networks, etc. Most of the literature has focused on the online model where a decision cannot be undone or on the streaming model where there is a specific memory budget not to be exceeded. However, as observed by Lattanzi and Vassilvitskii [1] the online model may appear too restrictive: if a bad decision has been made, it is sometime fine to spend some time to correct it instead of suffering the bad decision (i.e.: keeping a bad clustering) for the rest of the stream. However, spending too much time on the modification of the clustering may be counterproductive and that's what we aim at capturing in this model: keeping a good clustering by spending the least among of time and making as few changes to the current clustering as possible. For dynamic datasets, we do believe that the facility location formulation of the problem is very well suited since the 'ground truth number of clusters' of the underlying data may evolve and the facility location problem takes this into account through the facility cost.

Reviewer #3 on k-means/k-median algorithm: In Algorithm DeletePoint, notice that t_l is the latest point in time that MeyersonCapped (of the particular copy of the algorithm) placed a center. Since time t_l, more points might have been inserted for which we did not try to open a center as the number of centers would exceed the cap: this is the set of points $x_t, x_{t+1}, \ldots, x_{|X|}$ from which we try to open a new center (to reach the cap again, after the deletion) and recompute the assignment for the rest of the points.

Indeed, Algorithms MeyersonCapped and DeletePoint are executed on each of the $O(\log n)$ copies of the algorithm identified by index i, and hence, index i is not a good choice for iterating over the set of points $x_t, x_{t+1}, \ldots, x_{|X|}$. We will change that in the revised version of our paper. Moreover, we are going to fix all inconsistencies in the supplementary material.

References

[1] S. Lattanzi and S. Vassilvitskii. Consistent k-clustering. In *ICML*, pages 1975–1984, 2017.

[2] R. R. Mettu and C. G. Plaxton. Optimal time bounds for approximate clustering. *Mach. Learn.*, 56(1-3):35–60, June 2004.