We thank all the reviewers and ACs for their work in this challenging time. We will fix all typos found and improve the presentation of the appendix.

Constants in the approximation factor We agree with the reviewers that the constants in the statements of our theorems are not small. We stress that as in prior work (see [17], and [42]) we did not optimize for the worst case approximation constants. The bulk of the constant comes from the use of the well-known Meyerson sketch. Our results are very close to that of the simpler insertion-only case. Specifically, [42] shows a $2^{3p+5}\rho$ approximation for the insertion-only case (here p is the norm, ρ is the offline algorithm factor). This compares with our result in Lemma D.1 which is only a factor of 2 higher, $2^{3p+6}\rho$, but solves the more general problem. Note that the the SODA16 baseline [17] does not explicitly state the constant factor thus making a precise comparison difficult, but it uses the same Meyerson sketch as a subroutine thus incurring similar constants (see Lemma 3.2 in [17]). Importantly, these large factors appear only in the worst case analysis, and as our experiments (as well those of [42]) confirm that real instances behave much better. This could be explained by the fact that only a small fraction of all orderings of a set of points lead to large approximations in the Meyerson sketch (this has been shown formally by prior work).

Reviewer 1: For our algorithm we used a standard optimization from the literature ([42]): running one copy of the Meyerson sketch instead of the $O(\log(1/\gamma))$ copies that are needed for high probability statements. We also developed a lazy evaluation of the cost of the Meyerson sketch that saves update time. Notice that for fairness of comparison with SODA16 [17] (which uses at its core a Meyerson sketch as well) we use the same implementation of the sketch (with the same optimizations) for the two algorithms. This shows that our speedups over SODA16 come mostly from avoiding the additional factors of their algorithm; we will clarify this detail in the updated paper. We will also clarify in the $O()$ notation the dependency on p. We will improve the presentation in the appendix and we will change the notation of A_τ vs A_λ as suggested. *Q: Line 827: What is m?* A: It is the lower bound on the optimum as in the preliminary section. We will clarify that here we mean that, for the first threshold in Λ, the associated sketch A_m is not empty. *Q: Line 828: Well it is clear that $\lambda*$ exists.* A: We mean that it is in the set of the thresholds Λ for which we computed a sketch. *Q: Line 832: from the invariants.* A: We will clarify that it is invariant (iv) in Lemma D.2

Reviewer 2: Please see the answer above on the approximation factor. We will state the intuition behind the probability of adding a center to the Meryerson sketch. This is now a standard approach; the reason is that a point that is far from the current centers should be be added to avoid a large cost for that point. We ran experiments with k-median objective, $p = 1$; the results are in line with that of $p = 2$. For instance, for vmeasure accuracy over our datasets with ground-truth (see L.327 for the setup) the offline algorithm, our stream algorithm and the sampling baseline obtain 81.8%, 79.9%, 78.5%, respectively, confirming the same trend observed for k-means. We will add more details in the paper.

Reviewer 3: We will clarify at the beginning of the paper that the algorithm works on arbitrary metric spaces which we access only through distance function evaluations. We will increase the size of the plots. *Q: L 41-31 .. Euclidean or metric?* A: We will clarify the citations. k-median, k-means, and k-center are NP-Hard even in the Euclidean case but there are constant factor approximation algorithms for the general metric space case as well. *Q: Lower bounds on the space complexity* A: Correct, but it is trivial to show that at least k points must be stored to provide any approximation. *Q: L 185: $\widehat{f}$ was not defined* A: We apologize for the notation, $\widehat{f}$ is defined in that line as an approximation to the cost of the ϵ-consistent mapping that is computed by the algorithm *Q: Algorithm 1: why is it called "Update of Meyerson"?* A: we will rename it as it is confusing, it processes indeed the entire stream not a single update. *Q: Table 1: why not use ratios?* A: We will use ratios to make the table easier to read. We reported the ratio of the cost of our algorithm over the baseline as a percentage, i.e. 102% means that the cost is 1.02 times the baseline cost. *Q: L 287: cost means time?* Correct, we mean update time. *Q: Table 2: Max percentage is out of w?* A: Yes *Q: Why not list also a comparison of the cost (objective value)?* A: For lack of space it is in supplemental material, Table 7, L.939 *Q: L 301 and 326: which of the baseline algorithms?* A: We will clarify better, in L.301 we are evaluating SODA16 while in L.326 we are comparing with the offline K-Means++ baseline. *Q: L 325: W grows in table 2* A: We apologize. This is visible in the supplemental material Table 5, L.923.

Reviewer 4: Concerning the settings where the improvement from k^3 to k is significant, we would like to stress that in many industrial applications on large scale datasets the number of clusters can be quite large. We provided some examples in the introduction. Empirically we observe that that our speedups over the prior SODA16 work are an order of magnitude even for k as low as 4 in the COVERTYPE dataset. Moreover, our algorithm is significantly simpler than the previous one. *Q: 1. Lines 60-71* We will discuss the running time in the related work. *Q: what m and M are in Lemma 3.1.* We will clarify that they are the lower and upper bound on the cost of the optimum as defined in the preliminaries. *Q: Line 186: "Note that when M and ...". Did you mean M/m instead of M?* Correct, our bound depends on M/m being polynomial in w.