We thank all the reviewers for their comments.

Reviewer 1:

R does not contain enough statistics to estimate effective resistances. Effective resistances of m edges graph depends on both left and right singular space: it is the diagonal entries of matrix $E_{\mathcal{G}}^\top L_{\mathcal{G}}^\dagger E_{\mathcal{G}}$. We can approximate $L_{\mathcal{G}}^\dagger$ by R, but we need to also approximate the left singular space of $E_{\mathcal{G}}$ for the left and right multiplication in the above expression.

As for the complexity, the preprocessing time matches current state-of-the art result of Gupta et al. [28] as they also require solving a quadratic program of Alon-Naor (since we can also answer (S, T)-cut queries, it is only fair to compare with Gupta et al.; Blocki et al. only answer queries when $T = V \backslash S$). The total time to compute all τ_i for $1 \leq i \leq m$ is $O(n^3)$. Solving a semi-definite program takes $\mathsf{poly}(n)$ time, where the exact polynomial depends on whether we use interior point, ellipsoid method, or primal-dual approach of Arora-Kale. Rest of all the computations is subsumed by this run-time. However, now solving any (S, T)-cut query requires $O(\min |S|, |T|/\varepsilon^2)$ time instead of $O(n^2)$ time required by Gupta et al. while achieving better accuracy bound. On top of that, since we now work with sparse-graph, the run time of solving MAX-CUT and SPARSEST-CUT decreases significantly as the existing SDP based algorithms have a large polynomial dependence on the number of edges.

Reviewer 2:

Note that edges that have high leverage score are more likely to be retained in the graph (this is necessary for the utility/accuracy) but we also have plausible deniability for that edge, i.e., the edge could be in the output graph due to the overlaid complete graph. How to balance the two is the subtle part of setting the appropriate parameters which follows from analyzing the error bound. This is true for any differential privacy application, there is no absolute privacy. We necessarily have to “leak” some information (in a controlled manner) to get some utility out of the analysis. We discuss this briefly on lines 103-106, and lines 154-156.

Regarding the comment about " the tradeoff due to privacy, the privacy cost, cannot be understood in the current paper", the whole point of giving the error bound is to crisply characterize that tradeoff. What we show (refer to Table 1) is that if we require stronger privacy guarantee (by making privacy parameter α smaller), the upper bound on the error gets worse (as $1/\alpha$). This is a typical tradeoff in applications of differential privacy.

Reviewer 3:

We do not think that the reviewer has even tried to read the paper as all of our response amounts to basically providing pointers to the text in the paper. None of their comments support/justify the overly harsh evaluation. We urge the AC to intervene and politely request to not consider the comments of the reviewer. Please see more details below.

Motivation/significance: We discuss a clear application to machine learning on lines 281-300. In particular, we discuss how to extend our results for private manifold learning using Laplacian eigenmaps. More generally, as we argue in the paper, graph analysis finds application in many problems in data science and machine learning. We focus on graph sparsification as it is central to many graph analysis problems. This is clearly spelled out in the Introduction. More precisely, the opening paragraph motivates the need for private analysis on graphs on lines 10-17, lists numerous applications of graph analysis on lines 29-36, and finally discusses why graph sparsification plays a central role (on lines 44-47 and lines 62-67).

Algorithm: There is a whole subsection (Section 3.1 on lines 129-163) that details each and every step of the algorithm, and provides motivation and justification for each part. We find the comments by the reviewer as frivolous since answer to each of their questions is easily accessible in that part of the paper. The algorithm is very clearly described in the text.

General comment regarding empirical evaluation:

This is an algorithms+theory paper. We give a general framework for privatizing analysis on graph. There is no single application here, our results simultaneously apply to many problems. Besides as per the CFP, "Algorithmic contributions should have at least an illustration of how the algorithm might eventually materialize into a machine learning application." We give more than just illustrations, we give concrete applications as we discussed above and a complete result for manifold learning. We disagree with the Reviewer 3 about the scope of NeurIPS.

More importantly, the applications are well established and studied, we would be reproducing old experiments without adding any value. The point here is that the performance of the algorithms does not suffer much while guaranteeing privacy. Establishing privacy empirically is not straightforward and therefore, many papers in the privacy track are not accompanied by experiments.