First of all we would like to thank all the reviewers for their careful reading and constructive comments. Please see the responses below.

Reviewer #1: Adding a table summarizing the sample complexity results for the different oracle types is a great suggestion. We also thank the reviewer for pointing out the two relevant references, which we will add in the revised version.

Reviewer #2: **1)** Comp. Complexity: In general, the computational complexities of our algorithms are roughly of the order of $O(|\Omega| + |\mathcal{S}|^3)$, dominated by querying $|\Omega|$ random samples and applying a rank factorization on the gram matrix $\mathbf{A}_{\mathcal{S}}\mathbf{A}_{\mathcal{S}}^T$. We will add the computational complexities of each algorithm in the revised version.

2) Tightness of Bounds: Again, generally speaking, we can show that the scaling of the upper and lower bounds with respect to the various parameters (such as Δ, k, and n) is similar up to polylog factors. We will discuss the tightness of our IT lower and algorithmic upper bounds in the final version.

3) Matrix completion: Indeed, as was mentioned in the introduction, the sample complexity in our approach is smaller than those obtained using matrix completion when we deal with *unquantized* responses. For quantized responses, however, as far as we know, current matrix completion literature handle *dithered* quantized responses only (namely, adding continuous noise, e.g., Gaussian, before quantization). In our setting, we deal with (possibly noisy) quantized responses without any continuous dithering, and so matrix completion results cannot be used. In fact, without dithering matrix completion algorithms will fail on quantized data (see, Davenport, M. et. al., “1-Bit Matrix Completion"), as they do not exploit the discrete structure of the data, which is the main source for the success of our algorithms. Nonetheless, following the reviewer’s comment, for direct responses, we can add experiments comparing our results with matrix completion algorithms if desired.

4) Scale of α and worst vs. average: Depending on the dataset, the scaling of α in Theorem 3 w.r.t. (Δ, k, n) may vary widely. For example, in the non-overlapping case, $\alpha = k_{\min}/n \leq 1/k$, where $k_{\min}$ is the size of the smallest cluster. We can add this and other examples in the revised version. In the worst-case, a positive α could be as small as $1/n$ (unreasonable in real datasets), which implies a query complexity of $O(n^2)$. This is much higher than our average case results, as expected. We will add a detailed discussion on the comparison.

5) More experiments: We have recently conducted a few more experiments on “Delicious Bookmarks" and “Last.FM" datasets from `grouplens.org`, which further confirm our conclusions. We can add these results in the revised paper.

Reviewer #3: **1)** Tightness of Bounds: Please see the response to Rev. 2. In the revised version, we will add a table summarizing the scaling of these bounds.

2) Dependency on Δ: The dependency of the result in Theorem 5 on Δ is hidden in α, which may vary widely in (Δ, k, n). Note that α decreases as a function of Δ, which implies that the query complexity increases with Δ. For example, consider the example of 3 equally-sized clusters A, B and C. Suppose $\Delta = 1$ and in that case $|A \setminus B \cup C| = |A| = n/3$, implying that $\alpha = 1/3$. Now suppose that $\Delta = 2$. In this case $A \cap B$ and $A \cap C$ are non-empty and therefore $|A \setminus B \cup C| = |A| - |A \cap B| - |A \cap C| < n/3$, namely, α is less than $1/3$. As mentioned above, we plan to add several examples which will further clarify the behaviour of α.

3) Non-overlapping case: In the non-overlapping case, $\alpha = k_{\min}/n \leq 1/k$, where $k_{\min}$ is the size of the smallest cluster. This implies that the query complexity in the best scenario is $O(nk \log n)$, and thus there is no contradiction.

4) Generative models: Indeed, for the generative models case, the exponential dependency of the upper bounds on Δ is inherent, as the information-theoretic lower bounds suggest, and so it is neither an artifact of the proposed algorithm nor the way we upper bound its performance.

5) Role of $|\mathcal{S}|$: The theoretical value of $T = |\mathcal{S}|$ in Theorem 5 does work. But in practice we can take even smaller values for T than this theoretical value, and still guarantee recovery. This makes sense because Theorem 5 gives a sufficient condition on T. Of course, the smaller the size of $\mathcal{S}$ is, the sample & comp. complexities are smaller as well.

6) Unquantized responses: Algorithm 2 (`FindSimilarity`) is designed so that the guarantees hold under a specific stochastic assumption. More concisely, the necessary size of $\mathcal{S}$ is not defined for arbitrary real-world datasets. Note however, that the main objective in the first part of the algorithm is to select a number of elements so that the gram matrix is of full rank. Therefore, for a real-world dataset, we can always sample #clusters (k) elements randomly and make all pairwise queries and subsequently check if the gram matrix is of full rank. If it is not, we discard the elements and sample again, until we succeed. This is the reason why we chose 5 movies in the experiments (recall that the number of clusters is 5). Most real-world datasets are well behaved so that the number of trials required is actually quite low. We will add this discussion in the final version.